International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 31, No. 9 (2017) 1750028 (37 pages)

DOI: 10.1142/S0218001417500288

World Scientific
www.worldscientific.com

Approximation of Unknown Multivariate Probability Distributions by Using Mixtures of Product Components: A Tutorial

Jiří Grim

Institute of Information Theory and Automation of the Czech Academy of Sciences
P. O. Box 18, Pod Vodárenskou věží 4, CZ-18208 Prague 8, Czech Republic
grim@utia.cas.cz

Received 14 April 2014
Accepted 23 January 2017
Published 27 March 2017

In literature the references to EM estimation of product mixtures are not very frequent. The simplifying assumption of product components, e.g. diagonal covariance matrices in case of Gaussian mixtures, is usually considered only as a compromise because of some computational constraints or limited dataset. We have found that the product mixtures are rarely used intentionally as a preferable approximating tool. Probably, most practitioners do not "trust" the product components because of their formal similarity to "naive Bayes models." Another reason could be an unrecognized numerical instability of EM algorithm in multidimensional spaces. In this paper we recall that the product mixture model does not imply the assumption of independence of variables. It is even not restrictive if the number of components is large enough. In addition, the product components increase numerical stability of the standard EM algorithm, simplify the EM iterations and have some other important advantages. We discuss and explain the implementation details of EM algorithm and summarize our experience in estimating product mixtures. Finally we illustrate the wide applicability of product mixtures in pattern recognition and in other fields.

Keywords: Multivariate statistics; product mixtures; naive Bayes models; EM algorithm; pattern recognition; neural networks; expert systems; image analysis.

1. Introduction

The probabilistic description of data is known to be a powerful tool for solving many practical problems. Having estimated a multivariate probability distribution from data in a suitable analytic form, we can derive theoretically well-justified solutions in various fields like recognition, prediction, statistical modeling, neural networks, machine learning, image processing and others. Considering the statistical pattern recognition we assume Bayesian decision-making based on the class-conditional probability distributions. The initial decision information is usually contained in some multidimensional training data and therefore the key problem of statistical

pattern recognition is to estimate the underlying unknown class-conditional distributions or density functions from the available datasets.

In practice the real-life multivariate densities are nearly always multimodal without any simple parametric description. Essentially, there are two possible ways to estimate the multimodal density in a practically applicable analytic form: the nonparametric Parzen (kernel) estimates[56] or multivariate Gaussian mixtures. However, in the case of large multidimensional datasets the application of Parzen estimates becomes clumsy because the training data have to be kept in memory and the kernel function has to be optimally smoothed. In high-dimensional spaces the optimal smoothing of Parzen estimates is crucial but computationally demanding. In this respect the approximating mixtures are clearly preferable because the mixture model is more handy and all mixture parameters can be efficiently optimized in full generality by EM algorithm.

1.1. *Mixtures of product components*

In this paper we consider approximation of unknown multidimensional probability distributions by mixtures of components defined as products of univariate discrete distributions or density functions (briefly, product mixtures). We recall that the assumed mixtures of product components do not imply the independence of variables. On the contrary, we can prove that any discrete distribution can be expressed as a product mixture[18] and, in continuous case, with increasing number of components the Gaussian mixtures approach the asymptotic accuracy of the nonparametric Parzen estimates.[23] From the computational point of view, the simplifying assumption of diagonal covariance matrices is counterbalanced by the increased number of components. In other words, instead of trying to choose the correct number of components with the most suitable initial parameters (e.g. Refs. 6, 9, 38 and 52) we use sufficiently many simple components in a product form. In this sense the concept of product mixtures "moves" towards nonparametric kernel estimates while keeping the computational complexity in bounds. Simultaneously, the optimization of a large number of components facilitates the EM convergence.

The product components simplify the EM iterations, increase the numerical stability of EM algorithm and have some specific advantages as approximation tools (cf. Sec. 4). First of all, any marginal distributions are directly available by omitting superfluous terms in the products. Consequently, in prediction tasks, any conditional distributions are easily computed and the product mixtures can be estimated directly from incomplete data without replacing the missing values (Sec. 4.4). The product components support an important subspace (structural) modification of multidimensional mixtures (Sec. 4.1), which provides a theoretical background for the probabilistic neural networks (PNN, Sec. 5.3).

Nevertheless, despite their useful properties, the simplicity of product components might appear too restrictive in some cases. A natural idea would be to use general covariance matrices in Gaussian components, but they have to be inverted in each

EM iteration and, moreover, they become frequently ill-conditioned in high-dimensional spaces (cf. Sec. 3.4). One possible compromise would be to use mixtures of dependence trees. The Gaussian dependence tree density function[11] corresponds to a sampled covariance matrix and can explicitly describe the statistical relations of pairs of variables at the level of a single component. However, we have recently found[31] that, for binary data, the information gain implied by a large number of dependence tree components is decreasing in the course of EM iterations. In other words, the dependence tree mixtures spontaneously "degrade" to the more "advantageous" product mixtures in final stages of convergence. We recall in this connection that the proof of asymptotic properties of Parzen estimates also assumes the kernel function in a product form.[56]

1.2. *Related works*

In literature the references to estimation of product mixtures do not appear frequently. Usually the product components (e.g. diagonal covariance matrices in Gaussian densities) are considered only as a compromise enforced by computational constraints or by limited datasets.[49,50,68] According to our best knowledge the product mixtures are rarely used intentionally as a preferable approximating tool. The only exception are multivariate Bernoulli mixtures, because there is no alternative in case of binary vectors.

It appears that, in the past, the approximation power of product mixtures has been underestimated. The most probable reason is their formal similarity to the "naive Bayes models," which assume the class-conditional independence of variables. The resulting distribution is then a mixture of class-conditional product components but they are estimated separately from labeled training data. We recall that in product mixtures the independence assumption holds for each single component but not for the mixture as a whole. In this sense the widely used term "naive Bayes models" is incorrectly applied to product mixtures.[51]

Another more serious reason for a limited application of product mixtures could relate to possible numerical failing of the EM algorithm in multidimensional spaces. Even when using long-double variables, the evaluation of mixture components in high dimensions may partly underflow[8] without any visible influence on the convergence properties. Thus the final unsatisfactory results may appear as a consequence of poor learning properties of product mixtures (cf. Sec. 3.4 for more details). The related literature seems to provide indirect evidence in this respect since the references to multidimensional mixtures are rather rare (cf. Ref. 6 (dimension $N = 19$, real data), Ref. 64 ($N = 2$/real), Ref. 9 ($N = 5, 13, 18, 30, 47$/real), Ref. 65 ($N = 26$/real) and Ref. 48 ($N = 400$/binary)).

One of the few positive references is the paper of Lowd and Domingos[51] comparing the properties of product mixtures and Bayesian networks. In a series of experiments on 47 datasets from the UCI repository (dimension $N = 5$–618) they found that the "naive Bayes" mixtures learned by EM algorithm perform comparably with

Bayesian networks, but "they are orders of magnitude faster in inference." The average log-likelihoods on the benchmark datasets (as a measure of model accuracy) are comparable for both methods, but in two high-dimensional cases ($N = 618$/real) and ($N = 168$/real) the product mixtures clearly outperformed the Bayesian networks.

In this paper we refer widely to our results published in recent years. Most of the references relate to multidimensional problems in various areas like pattern recognition and probabilistic neural networks[12,16,19,25–27,32] (dimension $N = 1024$/binary), texture modeling[24] ($N = 400$–3600/real), texture evaluation[36] ($N = 400$–900, real), preprocessing of screening mammograms[34]($N = 145$/real), image forgery detection[39] ($N = 63$/real) and others.

The remainder of the paper is organized as follows. In Sec. 2 we suggest a novel simple proof of the monotonic property of EM algorithm. In Sec. 3 we discuss in detail the different implementation aspects of EM algorithm like weighted likelihood estimates (Sec. 3.1), choosing the number of components (Sec. 3.2) and initial parameters (Sec. 3.3) and show a simple way of avoiding the underflow problems of EM algorithm in high-dimensional spaces (Sec. 3.4). Section 4 refers to different modifications of product mixtures. We prove the monotonic property of the structural modification of EM algorithm (Sec. 4.1), Bernoulli- and Gaussian-product mixtures are the subjects of Secs. 4.2 and 4.3 and the missing data problem is discussed in Sec. 4.4. Section 5 illustrates the application possibilities of product mixtures.

2. EM Algorithm for Estimating Mixtures

Considering finite mixtures we assume the following linear combination of component distributions:

$$P(\boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m), \quad \boldsymbol{w} = (w_1, \ldots, w_M), \ \sum_{m \in \mathcal{M}} w_m = 1, \quad (1)$$

where $\boldsymbol{x} \in \mathcal{X}$ are discrete or real data vectors, $\boldsymbol{w}$ is a vector of probabilistic weights, $F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ are discrete component distributions or continuous densities with parameters $\boldsymbol{\theta}_m$:

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M\}, \quad \boldsymbol{\theta}_m = \{\theta_{m1}, \theta_{m2}, \ldots, \theta_{mN}\},$$

and $\mathcal{M} = \{1, \ldots, M\}$ is the component index set.

The problem of estimating mixtures was first posed by Carl Pearson (in 1894)[57] in connection with dissection of frequency curves, but only since the late 1960s has there been a widely applicable iterative scheme to compute the maximum-likelihood estimates of mixture parameters. Formally, given a set $\mathcal{S}$ of independent observations of the underlying N-dimensional random vector:

$$\mathcal{S} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots\}, \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T \in \mathcal{X}, \quad (2)$$

we can maximize the related log-likelihood function

$$L = L(\boldsymbol{w}, \boldsymbol{\Theta}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\Theta}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m) \right] \quad (3)$$

by means of EM algorithm.[5] In a general form, the EM iteration equations can be expressed as follows[10,23,31]:

$$q(m|\boldsymbol{x}) = \frac{w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)}{\sum_{j \in \mathcal{M}} w_j F(\boldsymbol{x}|\boldsymbol{\theta}_j)}, \quad w'_m = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}), \quad m \in \mathcal{M},\ \boldsymbol{x} \in \mathcal{S}, \quad (4)$$

$$Q_m(\boldsymbol{\theta}_m) = \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} \log F(\boldsymbol{x}|\boldsymbol{\theta}_m) \quad \left(w'_m|\mathcal{S}| = \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}) \right), \quad (5)$$

$$\boldsymbol{\theta}'_m = \arg\max_{\boldsymbol{\theta}_m} \{Q_m(\boldsymbol{\theta}_m)\} = \arg\max_{\boldsymbol{\theta}_m} \left\{ \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} \log F(\boldsymbol{x}|\boldsymbol{\theta}_m) \right\}, \quad (6)$$

where the prime denotes the new parameter values in each iteration. In Sec. 2.1 we give a simple proof that the general iteration scheme (4)–(6) produces a nondecreasing sequence of values of the maximized criterion (3).

In view of the implicit relation (6), the EM algorithm transforms the analytically hard maximization of the log-likelihood function (3) to the iterative maximization of the weighted log-likelihood functions $Q_m(\boldsymbol{\theta}_m)$. In this way, any new application of EM algorithm is reduced to the explicit solution of Eq. (6) which is usually available as a weighted analogy of the standard maximum-likelihood (m.-l.) estimate (cf. Sec. 3.1). For example, in case of Gaussian components with the component means $\boldsymbol{\mu}_m$ and covariance matrices $\boldsymbol{\Sigma}_m$:

$$F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{\sqrt{(2\pi)^N \det \boldsymbol{\Sigma}_m}} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_m) \right\}, \quad m \in \mathcal{M}, \quad (7)$$

we obtain the following explicit solution of Eq. (6):

$$\boldsymbol{\mu}'_m = \frac{1}{w'_m|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} \boldsymbol{x} q(m|\boldsymbol{x}), \quad m \in \mathcal{M}, \quad (8)$$

$$\boldsymbol{\Sigma}'_m = \frac{1}{w'_m|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} (\boldsymbol{x} - \boldsymbol{\mu}'_m)(\boldsymbol{x} - \boldsymbol{\mu}'_m)^T q(m|\boldsymbol{x}). \quad (9)$$

The standard reference to the EM algorithm is the paper of Dempster *et al.*,[5] at present one of the "all-time top 10" of statistics.[55] The paper introduces the name EM algorithm and shows its various application possibilities. The primary subject of the paper is the general problem of incomplete data; the estimation of mixtures is only included as one of the possible application areas (Ref. 5, pp. 15–17). With respect to mixtures, the EM iteration scheme has its own history (cf. e.g. the works of Dempster *et al.*[5] and Grim[10] for more details) which is closely related to the standard

likelihood equations for mixtures. There is no analytical solution of these equations but they can be rearranged to a form suggesting an iterative scheme. Hasselblad[42,43] is credited with first utilizing the computational advantages of this scheme — today known as the EM algorithm. According to Hosmer,[46] "Iterative m.-l. estimates" were proposed by Hasselblad and subsequently have been looked at by Day,[4] Hosmer[46] and Wolfe.[69] The recursive likelihood equations have been successfully applied to multivariate mixtures of different types. Several authors reported that the procedure increases the likelihood criterion at each iteration but they were not able to prove it.[58] The first proof of the key monotonic property of EM algorithm was published by Schlesinger[61,62] (cf. the work of Grim[10]), and further reported in a monograph of Ajvazjan *et al.*[1] and in a survey paper.[47] In the next subsection we reproduce Schlesinger's[61,62] idea in a novel simplified version.

2.1. *Monotonic property of EM algorithm*

We show that the sequence of log-likelihood values generated by the general iteration Equations (4)–(6) is nondecreasing in the sense of the inequality

$$L(\boldsymbol{w}', \boldsymbol{\Theta}') \geq L(\boldsymbol{w}, \boldsymbol{\Theta}). \tag{10}$$

Recall first that the Kullback–Leibler information divergence is nonnegative[66] for any two discrete probability distributions q_m, q'_m:

$$I(q||q') = \sum_{m\in\mathcal{M}} q_m \log \frac{q_m}{q'_m} \geq 0, \quad q_m, q'_m \geq 0, \quad \sum_{m\in\mathcal{M}} q_m = \sum_{m\in\mathcal{M}} q'_m = 1, \tag{11}$$

and equals zero if and only if $q_m = q'_m$ for all $m \in \mathcal{M}$. Note that the above inequality can be rewritten in the following equivalent form:

$$\sum_{m\in\mathcal{M}} q_m \log q'_m \leq \sum_{m\in\mathcal{M}} q_m \log q_m, \tag{12}$$

which implies that the left-hand side of (12), as a function of $q'_1, q'_2, \ldots, q'_M$, is uniquely maximized by $q'_m = q_m, m \in \mathcal{M}$.

In view of the general inequality (11) we can write

$$\frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} \left[\sum_{m\in\mathcal{M}} q(m|\boldsymbol{x}) \log \frac{q(m|\boldsymbol{x})}{q'(m|\boldsymbol{x})} \right] \geq 0. \tag{13}$$

By substitution for $q(m|\boldsymbol{x})$ and $q'(m|\boldsymbol{x})$ from (4) we obtain

$$\frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} \sum_{m\in\mathcal{M}} q(m|\boldsymbol{x}) \log \frac{P'(\boldsymbol{x})}{P(\boldsymbol{x})} - \frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} \sum_{m\in\mathcal{M}} q(m|\boldsymbol{x}) \log \left[\frac{w'_m F(\boldsymbol{x}|\boldsymbol{\theta}'_m)}{w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)} \right] \geq 0, \tag{14}$$

where the first sum on the left-hand side is the log-likelihood increment

$$\frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} \left[\sum_{m\in\mathcal{M}} q(m|\boldsymbol{x}) \right] \log \frac{P'(\boldsymbol{x})}{P(\boldsymbol{x})} = \frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} \log \frac{P'(\boldsymbol{x})}{P(\boldsymbol{x})} = L' - L \tag{15}$$

and therefore we can write

$$L' - L \geq \sum_{m\in\mathcal{M}} \left[\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} q(m|\boldsymbol{x})\right] \log\frac{w'_m}{w_m} + \sum_{m\in\mathcal{M}} \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} q(m|\boldsymbol{x}) \log\frac{F(\boldsymbol{x}|\boldsymbol{\theta}'_m)}{F(\boldsymbol{x}|\boldsymbol{\theta}_m)}. \quad (16)$$

Considering the substitutions (4) and the inequality (11), we can write

$$\sum_{m\in\mathcal{M}} \left[\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} q(m|\boldsymbol{x})\right] \log\frac{w'_m}{w_m} = \sum_{m\in\mathcal{M}} w'_m \log\frac{w'_m}{w_m} \geq 0, \quad (17)$$

and therefore the first sum on the right-hand side of the inequality (16) is nonnegative. Further, by definition (6), we can write for any parameters $\boldsymbol{\theta}_m$:

$$\sum_{x\in\mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} \log F(\boldsymbol{x}|\boldsymbol{\theta}'_m) \geq \sum_{x\in\mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} \log F(\boldsymbol{x}|\boldsymbol{\theta}_m), \quad m \in \mathcal{M}. \quad (18)$$

By summing and rearranging the above inequalities (18) we obtain

$$\sum_{m\in\mathcal{M}} \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} q(m|\boldsymbol{x}) \log\frac{F(\boldsymbol{x}|\boldsymbol{\theta}'_m)}{F(\boldsymbol{x}|\boldsymbol{\theta}_m)} \geq 0. \quad (19)$$

Consequently, in view of the inequalities (17) and (19), the increment of the log-likelihood criterion is nonnegative:

$$L' - L \geq \sum_{m\in\mathcal{M}} w'_m \log\frac{w'_m}{w_m} + \sum_{m\in\mathcal{M}} \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} q(m|\boldsymbol{x}) \log\frac{F(\boldsymbol{x}|\boldsymbol{\theta}'_m)}{F(\boldsymbol{x}|\boldsymbol{\theta}_m)} \geq 0, \quad (20)$$

and the sequence of the log-likelihood values is nondecreasing in the sense of inequality (10).

The inequality (20) implies that, if the criterion (3) is bounded above, the sequence of log-likelihood values converges, possibly to a local maximum (or a saddle point) of the log-likelihood function $L(\boldsymbol{w}, \boldsymbol{\Theta})$. We finally remark that, for the sake of the proof, we only need to guarantee the inequality (18) which is not as strong as the maximization condition (6). This modification is called the generalized EM algorithm (GEM[5]).

Let us note that, in order to prove the monotonic property of EM algorithm, we do not introduce any additional "latent" variables, there is no need of any statistical interpretation of EM iteration scheme. Thus in case of any new modification of EM algorithm (cf. Sec. 4.1) it is only necessary to derive the new iteration equations in order to guarantee the basic inequality (19).

3. Implementation of EM Algorithm

For the sake of an efficient implementation of EM algorithm we need an explicit solution of Eq. (6) which is often available as a weighted analogy of the standard maximum-likelihood estimate.[10,23]

3.1. *Weighted likelihood estimate*

Let $F(\boldsymbol{x}|\boldsymbol{\theta})$ be a probability density (or discrete probability distribution) with the corresponding log-likelihood criterion

$$L(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log\ F(\boldsymbol{x}|\boldsymbol{\theta}). \tag{21}$$

Furthermore, let $\boldsymbol{\theta}^*$ be the maximum-likelihood estimate of $\boldsymbol{\theta}$ defined as an additive function of $\boldsymbol{x} \in \mathcal{S}$:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left\{ \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log\ F(\boldsymbol{x}|\boldsymbol{\theta}) \right\} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \boldsymbol{b}(\boldsymbol{x}). \tag{22}$$

Denoting by $N(\boldsymbol{x})$ the frequency of $\boldsymbol{x}$ in the data sample $\mathcal{S}$, we can equivalently rewrite Eqs. (21) and (22) in the form:

$$L(\boldsymbol{\theta}) = \sum_{x \in \tilde{\mathcal{X}}} \gamma(\boldsymbol{x}) \log F(\boldsymbol{x}|\boldsymbol{\theta}), \quad \gamma(\boldsymbol{x}) = N(\boldsymbol{x})/|\mathcal{S}|,\ \tilde{\mathcal{X}} = \mathcal{X} \cap \mathcal{S}, \tag{23}$$

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{x \in \tilde{\mathcal{X}}} \gamma(\boldsymbol{x}) \log F(\boldsymbol{x}|\boldsymbol{\theta}) \right\} = \sum_{x \in \tilde{\mathcal{X}}} \gamma(\boldsymbol{x}) \boldsymbol{b}(\boldsymbol{x}), \tag{24}$$

where $\gamma(\boldsymbol{x})$ stands for the relative frequency of $\boldsymbol{x}$ in $\mathcal{S}$. Note that Eq. (24) is a weighted version of (22). In view of this analogy an arbitrarily weighted log-likelihood function (23) is maximized by the respective weighted sum (24), (cf. (8) and (9)), provided that the formula (22) is available (cf. Theorem 5.1 of Ref. 10 for more details).

Let us simultaneously remark that the relative frequency notation (23) is also applicable to the log-likelihood function (3):

$$L(\boldsymbol{\theta}) = \sum_{x \in \tilde{\mathcal{X}}} \gamma(\boldsymbol{x}) \log \left[\sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m) \right] \tag{25}$$

and to the related EM iteration equations (4) and (6):

$$w'_m = \sum_{x \in \tilde{\mathcal{X}}} \gamma(\boldsymbol{x}) q(m|\boldsymbol{x}), \tag{26}$$

$$\boldsymbol{\theta}'_m = \arg\max_{\boldsymbol{\theta}_m} \left\{ \sum_{x \in \tilde{\mathcal{X}}} \frac{\gamma(\boldsymbol{x}) q(m|\boldsymbol{x})}{w'_m} \log F(\boldsymbol{x}|\boldsymbol{\theta}_m) \right\}. \tag{27}$$

In this way the EM algorithm can be applied to arbitrarily weighted data, e.g. to utilize some external knowledge about the meaning or relevance of data. In a case of discrete distributions defined by tables, we can compute an equivalent mixture representation by setting $\mathcal{S} \equiv \mathcal{X}$ and $\gamma(\boldsymbol{x})$ equal to the respective table values.[11]

3.2. *Estimation versus approximation*

The statistical problem of identification of mixtures (as in cluster analysis[18]) assumes that the solution is unique. For this reason the mixtures have to be identifiable and we have to specify the number of components in the mixture correctly in advance. Obviously, by fitting e.g. a mixture of three components to a dataset containing four well-separated clusters, we have several different alternatives which may correspond to different local maxima of the log-likelihood function. In view of possible local maxima, the EM algorithm depends on its starting point, and the initial mixture parameters should therefore be reasonably chosen. In literature there are extensive discussions about initializing mixtures and a proper choice of the number of mixture components,[9,18,52–55] but they are less relevant in the case of approximation problems.

Recall that, with the increasing dataset $\mathcal{S}$, the log-likelihood criterion can be viewed as an estimate of the following asymptotic expectation with respect to the unknown true probability distribution P^*:

$$L = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log P(\boldsymbol{x}) \rightarrow E_{P^*}\{\log P(\boldsymbol{x})\}. \tag{28}$$

By maximizing the last expression we minimize Kullback–Leibler divergence

$$I(P^*, P) = E_{P^*}\left\{\log \frac{P^*(\boldsymbol{x})}{P(\boldsymbol{x})}\right\} = E_{P^*}\{\log P^*(\boldsymbol{x})\} - E_{P^*}\{P(\boldsymbol{x})\}, \tag{29}$$

which can be viewed as a dissimilarity measure between the true unknown probability distribution $P^*(\boldsymbol{x})$ and the estimated mixture $P(\boldsymbol{x})$. In this sense the maximum-likelihood criterion is applicable from the point of view of approximation.

In this paper we use the term approximating mixture to emphasize that the approximation accuracy is of primary importance. This theoretical detail has important consequences since, unlike estimation problems, in approximation problems with a large number of components ($M \approx 10^1$–10^2) the situation is rather different. First, the approximating mixture need not be identifiable. On the contrary, non-identifiable mixtures are more flexible and less prone to be "trapped" in a local optimum. Concerning the mixture complexity, the EM algorithm is known to have a tendency to suppress the weights of superfluous components, but this mechanism is not strong enough to control the mixture complexity. Nevertheless, the resulting distribution of component weights is typically sigmoidal and therefore the mixture contains a large portion of components having very low weights. As these components can be omitted without any significant consequences, the exact initial number of components usually does not play essential role in multidimensional approximation problems.

A large number of components has a positive aspect from the point of view of "overfitting" since the EM algorithm automatically decreases the influence of outliers which are typically "covered" by a single component with a low weight. As a

result, the product mixtures are relatively robust with respect to overfitting. In the experiments[26,27] we have observed that the optimal mixture model may include hundreds of thousands of parameters without any relevant loss of generalizing properties.

3.3. *Initial parameters*

The optimal number of components is closely related to the problem of initial parameters. There are many different possibilities to fit a mixture of many components to a large number of multidimensional measurements but the related local maxima of the maximum-likelihood function usually do not differ very much and, in view of Eqs. (28) and (29), the corresponding approximation quality is comparable. In this sense the choice of initial parameters does not play an essential role. For the same reason in approximation tasks the frequently discussed slow convergence in final iterations can be simply interrupted by a relative increment threshold.

In our experience, the product mixtures can be initialized randomly in many approximation problems (cf. Sec. 5.1). The initial weights may be uniform and the initial component parameters may be specified randomly near the "global" means. In this way, we can suppress undesirable interference of component "trajectories" with the course of iterations. In Gaussian components the variances may be identical but they should be sufficiently large to make all data "visible" to all components at the beginning. In the case of complex highly multimodal data it may be useful to initialize the component means by randomly chosen training data vectors — in a way resembling the Parzen estimates in a certain sense.

In high-dimensional spaces the competitive suppression of component weights may become too strong and some components may be suppressed before they succeed to reach a "stable" position. Consequently, in some cases it may be difficult "to keep the components in game." A useful idea is to fix the uniform component weights in several starting EM iterations. Note that this approach is similar to the popular nearest mean algorithm which implicitly assumes the uniform component weights.

It is intuitively clear that all the above suggestions in Secs. 3.2 and 3.3 are data-dependent and therefore they cannot be generally justified in a rigorous way. Nonetheless, they can be helpful in solving practical problems.

3.4. *Multidimensional mixtures*

Multidimensional spaces are known to be "sparse" in the sense that the distances between data points become large due to high dimensionality. For this reason the multidimensional components $w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ are often nearly nonoverlapping. Intuitively it is clear that a measure of component overlap may contain useful information about the estimated mixture. Thus the nonoverlapping components imply nearly binary properties of the conditional weights $q(m|\boldsymbol{x})$. In other words, for a given $\boldsymbol{x}$ there is usually one component with a high conditional weight $q(m|\boldsymbol{x}) \approx 1$ whereas

the remaining conditional weights are close to zero. In this respect the following mean value of the maximum conditional weights

$$\bar{q}_{\max} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} q_{\max}(\boldsymbol{x}), \quad q_{\max}(\boldsymbol{x}) = \max_{m \in \mathcal{M}} \{q(m|\boldsymbol{x})\}, \tag{30}$$

provides a useful insight into the properties of the underlying mixture estimate and may be helpful to balance the relationship between the problem dimension and model complexity.

In numerical experiments with a large number of components ($M \approx 10^1$–10^2) in high-dimensional spaces ($N \approx 10^2$–10^3) we usually obtain relatively large values $\bar{q}_{\max}$ in final iterations, typically $\bar{q}_{\max} \approx 0.975$–$0.999$.[24,34,41] We can conclude that, in such a case, for most data vectors $\boldsymbol{x} \in \mathcal{S}$ there is only one "dominating" component characterized by the conditional weight near to one ($q(m|\boldsymbol{x}) \doteq 1$) and, in turn, for each component $w_m F(\boldsymbol{x}|m)$ there is a corresponding subset of "dominated" data vectors $\boldsymbol{x} \in \mathcal{S}^{(m)} \subset \mathcal{S}$ with the same property. Recall now that, considering a general normal mixture, we compute in each EM iteration the new covariance matrices Σ'_m by Eq. (9) but the summing is actually reduced to the subsets $\mathcal{S}^{(m)}$:

$$\Sigma'_m = \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} (\boldsymbol{x} - \boldsymbol{\mu}'_m)(\boldsymbol{x} - \boldsymbol{\mu}'_m)^T \approx \sum_{x \in \mathcal{S}^{(m)}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} (\boldsymbol{x} - \boldsymbol{\mu}'_m)(\boldsymbol{x} - \boldsymbol{\mu}'_m)^T. \tag{31}$$

For a small weight w_m the subset $\mathcal{S}^{(m)}$ may become too small ($|\mathcal{S}^{(m)}| \doteq w_m|\mathcal{S}| < N$) to get a full-rank covariance matrix $\boldsymbol{\Sigma}_m$. Consequently, in the case of a large number of multidimensional normal components we frequently obtain ill-conditioned (nearly singular) covariance matrices in EM iterations which may cause numerical difficulties in matrix inversion. As this problem is inherent in multidimensional Gaussian mixtures with many components, it can be viewed as a computational argument for using diagonal covariance matrices.

In the case of high-dimensional real spaces, say for $N > 50$, the Gaussian components $F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ may yield very small values, often below the usual representation bounds. In this way some of the components may underflow, even when using variables of the long-double type. The small values are lost and cannot be "recovered" by norming in Eq. (4). The problem is obviously data-dependent, may occur in binary- or discrete-data spaces and may go unrecognized since the EM algorithm need not necessarily collapse. Thus, possible unsatisfactory results may be interpreted as an instance of "poor learning". Surprisingly, any related references in the literature are rare,[8] but the relatively low dimensionality of published numerical examples (cf. Sec. 1.2) seems to indirectly support this hypothesis.

The limited numerical accuracy of a processor can be easily bypassed since the computation of $q(m|\boldsymbol{x})$ in Eq. (4) is invariant with respect to multiplication of the components $w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ by arbitrary constant. Evaluating the components in

logarithmic form we simply find the corresponding maximum value

$$\log C_0(\boldsymbol{x}) = \max_m \{\log[w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)]\} \tag{32}$$

which can be used to compute the "normalized" values of $w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ and $P(\boldsymbol{x})$:

$$1 \geq C_0^{-1}(\boldsymbol{x}) w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m) = \exp\{-\log C_0(\boldsymbol{x}) + \log w_m + \log F(\boldsymbol{x}|\boldsymbol{\theta}_m)\}. \tag{33}$$

It can be seen that the constant $C_0(\boldsymbol{x})$ is reduced in formula (4) for $q(m|\boldsymbol{x})$ and the related log-likelihood values can be computed by the equation

$$\log P(\boldsymbol{x}) = \log \left[C_0^{-1}(\boldsymbol{x}) \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m) \right] + \log C_0(\boldsymbol{x}). \tag{34}$$

Obviously, the "normalized" component values (33) may underflow too, but the maximum of the expression $C_0^{-1}(\boldsymbol{x}) w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ is equal to one. Therefore the small component values may be neglected without any relevant loss of accuracy in Eqs. (4) and (5).

4. Product Mixtures and Their Modifications

Considering mixtures of product components, we approximate the unknown probability distributions in the form of the following conditional independence model

$$P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\Theta}) = \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m), \quad \boldsymbol{x} \in \mathcal{X},\ \mathcal{N} = \{1, \ldots, N\}, \tag{35}$$

$$F(\boldsymbol{x}|\boldsymbol{\theta}_m) = \prod_{n \in \mathcal{N}} f_n(x_n|\theta_{mn}), \quad \boldsymbol{\theta}_m = \{\theta_{m1}, \theta_{m2}, \ldots, \theta_{mN}\}, \tag{36}$$

where $\boldsymbol{x} \in \mathcal{X}$ are discrete or real data vectors, $\boldsymbol{w}$ is the vector of probabilistic weights and the component distributions $F(\boldsymbol{x}|\boldsymbol{\theta}_m)$ with the parameters $\boldsymbol{\theta}_m$ are defined as products. Here $f_n(x_n|\theta_{mn})$ are univariate discrete probability distributions or density functions with the parameters θ_{mn} and $\mathcal{N}$ is the index set of variables.

Let us remark that, in the case of product components, Eq. (5) can be specified for each variable independently ($n \in \mathcal{N}, m \in \mathcal{M}$):

$$Q_{mn}(\theta_{mn}) = \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m |\mathcal{S}|} \log\ f_n(x_n|\theta_{mn}), \quad \theta'_{mn} = \arg\max_{\theta_{mn}} \{Q_{mn}(\theta_{mn})\}. \tag{37}$$

As mentioned earlier, the product mixture does not imply the statistical independence of the variables (cf. Sec. 1.2) and therefore the term "naive Bayes models" is incorrectly applied to product mixtures. It can be easily verified that in case of discrete variables the product mixtures are universal approximators in the sense that any discrete distribution can be expressed as a product mixture (cf. Ref. 18, Remark 1, p. 643). In the case of continuous variables the Gaussian product mixtures approach the universality of nonparametric Parzen estimates[56] with the increasing number of components. Simultaneously, the diagonal covariance matrices in

Gaussian components avoid matrix inversions in EM iterations and increase the computational stability of EM algorithm (cf. Secs. 1.2 and 3.4). In the following subsections we show that, in addition, the mixtures of product components have some other useful properties.

4.1. *Structural mixture model*

One of the most important properties of product mixtures is the possibility of structural modification. The structural (subspace) mixture model has been proposed within the framework of statistical pattern recognition[12] as a more general alternative to feature selection. In particular, making the substitution

$$F(\boldsymbol{x}|\boldsymbol{\theta}_m) = F(\boldsymbol{x}|\boldsymbol{\theta}_0)G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m), \quad m \in \mathcal{M}, \tag{38}$$

in (35), we introduce the structural mixture

$$P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\phi}) = F(\boldsymbol{x}|\boldsymbol{\theta}_0) \sum_{m\in\mathcal{M}} w_m G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m), \tag{39}$$

where $F(\boldsymbol{x}|\boldsymbol{\theta}_0)$ is a fixed nonzero "background" probability distribution, and the component functions $G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)$ include binary structural parameters $\phi_{mn} \in \{0, 1\}$:

$$G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) = \prod_{n\in\mathcal{N}} \left[\frac{f_n(x_n|\theta_{mn})}{f_n(x_n|\theta_{0n})}\right]^{\phi_{mn}}, \quad \boldsymbol{\phi}_m = (\phi_{m1}, \ldots, \phi_{mN}) \in \{0, 1\}^N. \tag{40}$$

A convenient fixed choice of the background distribution is the product of the global univariate marginals[a]:

$$F(\boldsymbol{x}|\boldsymbol{\theta}_0) = \prod_{n\in\mathcal{N}} f_n(x_n|\theta_{0n}), \quad f_n(x_n|\theta_{0n}) \approx P_n^*(x_n), \ n \in \mathcal{N}.$$

Setting a structural parameter $\phi_{mn} = 0$, we can replace any component-specific distribution $f_n(x_n|\theta_{mn})$ by its respective background distribution $f_n(x_n|\theta_{0n})$, i.e. we can write

$$F(\boldsymbol{x}|\boldsymbol{\theta}_m) = \prod_{n\in\mathcal{N}} f_n(x_n|\theta_{mn})^{\phi_{mn}} f_n(x_n|\theta_{0n})^{1-\phi_{mn}}. \tag{41}$$

Obviously, Eq. (41) and the structural mixture model in (38) and (39) can be viewed formally as a product mixture again.

Let us note that the component functions $G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)$ may be defined on different subspaces. The number of involved parameters and the "structure" of the finite mixture (39) can be controlled by means of the binary parameters ϕ_{mn}. In other words, we can estimate product mixtures of high dimensionality while keeping the number of estimated parameters reasonably small, e.g. appropriate to the size of the available dataset. Simultaneously, by including only the most relevant specific

[a] In a more general version the background distribution can be optimized too.[12]

parameters into components, we suppress the influence of the less informative "noisy" variables.

The structural mixture (39) can be optimized in full generality, including the structural parameters ϕ_{mn}, by means of EM algorithm[12,16,24,32]:

$$L = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_0) G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) \right].$$

Again, in the following general EM iteration equations, the prime denotes the new parameter values ($m \in \mathcal{M}, n \in \mathcal{N}, \boldsymbol{x} \in \mathcal{S}$):

$$q(m|\boldsymbol{x}) = \frac{w_m G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}{\sum_{j \in \mathcal{M}} w_j G(\boldsymbol{x}|\boldsymbol{\theta}_j, \boldsymbol{\phi}_j)}, \quad w'_m = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}), \tag{42}$$

$$Q_{mn}(\theta_{mn}) = \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m |\mathcal{S}|} \log f_n(x_n|\theta_{mn}) \quad \left(w'_m |\mathcal{S}| = \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}) \right), \tag{43}$$

$$\theta'_{mn} = \arg\max_{\theta_{mn}} \{Q_{mn}(\theta_{mn})\} = \arg\max_{\theta_{mn}} \left\{ \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m |\mathcal{S}|} \log f_n(x_n|\theta_{mn}) \right\}, \tag{44}$$

and γ'_{mn} is the structural criterion given by

$$\gamma'_{mn} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}) \log \frac{f_n(x_n|\theta'_{mn})}{f_n(x_n|\theta_{0n})}, \quad m \in \mathcal{M},\ n \in \mathcal{N}. \tag{45}$$

Assuming a fixed number of component-specific parameters,

$$\sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \phi'_{mn} = \lambda, \tag{46}$$

we define the optimal subset of nonzero parameters ϕ'_{mn} by choosing the largest λ values such that $\gamma'_{mn} > 0$.

It can be verified[12,16] that the iteration scheme (42)–(46) guarantees the monotonic property of the EM algorithm. Recall that, for this purpose, we cannot refer to the general proof given in Sec. 2 because the optimization of the structural parameters ϕ_{mn} depends on all other parameters. Instead we have to prove that the basic inequality (20) follows from the above Eqs. (42)–(46). In particular, in view of Eqs. (20) and (38), it is sufficient to prove the inequality

$$\sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}) \log \frac{G(\boldsymbol{x}|\boldsymbol{\theta}'_m, \boldsymbol{\phi}'_m)}{G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)} \geq 0, \tag{47}$$

which can be rewritten as follows (cf. (40)):

$$\sum_{m \in \mathcal{M}} \sum_{x \in \mathcal{S}} \frac{q(m|\boldsymbol{x})}{|\mathcal{S}|} \sum_{n \in \mathcal{N}} \left[\phi'_{mn} \log \frac{f_n(x_n|\theta'_{mn})}{f_n(x_n|\theta_{0n})} - \phi_{mn} \log \frac{f_n(x_n|\theta_{mn})}{f_n(x_n|\theta_{0n})} \right] \geq 0,$$

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\sum_{x\in\mathcal{S}}\frac{q(m|\boldsymbol{x})}{|\mathcal{S}|}\left[(\phi'_{mn}-\phi_{mn})\log\frac{f_n(x_n|\theta'_{mn})}{f_n(x_n|\theta_{0n})}+\phi_{mn}\log\frac{f_n(x_n|\theta'_{mn})}{f_n(x_n|\theta_{mn})}\right]\geq 0.$$

Considering the definition (45) of the structural criterion γ'_{mn}, we can equivalently write the last inequality in the form:

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}(\phi'_{mn}-\phi_{mn})\gamma'_{mn}+\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\phi_{mn}\sum_{x\in\mathcal{S}}\frac{q(m|\boldsymbol{x})}{|\mathcal{S}|}\log\frac{f_n(x_n|\theta'_{mn})}{f_n(x_n|\theta_{mn})}\geq 0. \tag{48}$$

Recall that the definition (44) implies the inequalities ($n\in\mathcal{N},\ m\in\mathcal{M}$):

$$\sum_{x\in\mathcal{S}}\frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|}\log f_n(x_n|\theta'_{mn})\geq\sum_{x\in\mathcal{S}}\frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|}\log f_n(x_n|\theta_{mn}),\quad\forall\,\theta_{mn}, \tag{49}$$

and consequently, for any fixed structural parameters ϕ_{mn}, we can write

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\phi_{mn}\sum_{x\in\mathcal{S}}\frac{q(m|\boldsymbol{x})}{|\mathcal{S}|}\log\frac{f_n(x_n|\theta'_{mn})}{f_n(x_n|\theta_{mn})}\geq 0,\quad\forall\,\theta_{mn}. \tag{50}$$

For the same reason the structural criterion γ'_{mn} is nonnegative and therefore

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\phi'_{mn}\gamma'_{mn}\geq\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\phi_{mn}\gamma'_{mn} \tag{51}$$

because the binary structural variables ϕ'_{mn} are nonzero only for the λ largest values of γ'_{mn} (cf. (46)). Thus, in view of the last two inequalities (50) and (51) the inequality (48) as well as (47) is valid and therefore the monotonicity condition (20) is guaranteed for the structural modification of EM algorithm (42)–(46).

Let us remark that, in Eq. (46), we can fix the number of specific parameters in each component individually, i.e. we can write

$$\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\phi'_{mn}=\lambda_m,\quad m\in\mathcal{M}. \tag{52}$$

For this case the above proof can be easily modified by omitting the sum for $m\in\mathcal{M}$ in Eq. (47).

Note that the reduced background probability distribution $F(\boldsymbol{x}|\boldsymbol{\theta}_0)$ in the formula (42) makes the EM iterations more efficient. On the other hand, in each iteration, the values of the structural criterion γ'_{mn} are always evaluated for all $m\in\mathcal{M}, n\in\mathcal{N}$ and ordered. From the computational point of view, it is therefore more convenient to specify the structural parameters by simple thresholding:

$$\phi'_{mn}=\begin{cases}1, & \gamma'_{mn}>\tau\\ 0, & \gamma'_{mn}\leq\tau\end{cases}\quad\left(\tau=\frac{\rho}{MN}\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{N}}\gamma'_{mn},\quad\rho\geq 0\right), \tag{53}$$

The monotonic property seems to hold in this case too, but the above proof is not more applicable.

The structural mixture model can be viewed as a more general alternative to standard feature-selection techniques. By estimating the structural parameters we are not confined to a single subset of features. The computation of class-conditional distributions may be reduced to optimal subsets of the most informative variables which are specific for each component. In other words we obtain a specific subset of features for each component. Simultaneously, the structural model is less prone to overfitting because in the components the unused variables are usually noisy and unreliable.[27]

The structural model (39) can be used to solve the feature-selection problem globally for multimodal densities,[59] (see also Ref. 12, Remark 4.2, p. 150). Recently a similar model has apparently been independently proposed by Graham and Miller[9] to control the mixture model complexity in combination with the minimum description length criterion (see also Markley and Miller[52] and Bouguila *et al.*[3]). We remark in this connection that the structural mixture alone includes a specific internal mechanism to reduce the mixture complexity. As a consequence of structural optimization, some components can "lose" all specific parameters and the resulting identical "nonspecific" components can be replaced by a single one. In this way the final number of components may decrease according to a suitably chosen parameter ρ in Eq. (53).

In statistical pattern recognition the "structural" computation can be viewed as "dimension-independent" because any variable x_n which is nonspecific in all components (i.e. $\phi_{mn} = 0$ for all $m \in \mathcal{M}$) can be omitted and, in turn, we can include any new variable x_{N+1} given the corresponding background distribution.[22]

4.2. *Mixtures of multivariate Bernoulli distributions*

In the case of binary data the product mixture model is known as a multivariate Bernoulli mixture (cf. Refs. 19, 25–27, 32 and 60) having the components

$$F(\boldsymbol{x}|\boldsymbol{\theta}_m) = \prod_{n\in\mathcal{N}} p_n(x_n|\theta_{mn}), \quad x_n \in \{0,1\}, \tag{54}$$

with the univariate distributions defined by the equation

$$p_n(x_n|\theta_{mn}) = (\theta_{mn})^{x_n}(1-\theta_{mn})^{1-x_n}, \quad 0 \le \theta_{mn} \le 1.$$

Here and in equations that follow, we use the notation $p_n(x_n|\cdot)$ instead of $f_n(x_n|\cdot)$ to emphasize that the univariate distributions are discrete.

Note that the maximization task (6) can be solved separately (cf. (37)) for each parameter θ_{mn} and, considering Eq. (24), we can write ($n \in \mathcal{N},\ m \in \mathcal{M}$):

$$\theta'_{mn} = \arg\max_{\theta_{mn}} \left\{ \sum_{x\in\mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} \log p_n(x_n|\theta_{mn}) \right\} = \sum_{x\in\mathcal{S}} x_n \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|}. \tag{55}$$

The EM algorithm for estimating Bernoulli mixtures can be easily transformed to the structural version of Sec. 4.1 by introducing the binary structural variables ϕ_{mn}.

It can be seen that the corresponding structural criterion γ'_{mn} (cf. (45)) can be rewritten in the form:

$$\gamma'_{mn} = w'_m \sum_{\xi=0}^{1} p'_n(\xi|m) \log \frac{p'_n(\xi|m)}{p_n(\xi|0)} = w'_m I(p'_n(\cdot|m)||p_n(\cdot|0)). \tag{56}$$

In other words, the structural criterion γ'_{mn} can be expressed in terms of Kullback–Leibler information divergence $I(p'_n(\cdot|m)||p_n(\cdot|0))$ between the component-specific distribution $p'_n(x_n|m)$ and the corresponding univariate "background" distribution $p_n(x_n|0)$. Thus, only the most distinct (i.e. specific and informative) distributions $p'_n(x_n|m)$ are included in the components.

4.3. *Gaussian mixtures of product components*

In order to approximate continuous multivariate and multimodal densities we can use mixtures of Gaussian components with diagonal covariance matrices (cf. Refs. 20, 24, 34, 36, 39 and 41):

$$\begin{aligned} P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \sum_{m\in\mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m), \quad \boldsymbol{x} \in \mathcal{R}^N, \\ \boldsymbol{\mu}_m &= (\mu_{m1}, \ldots, \mu_{mN}), \quad \boldsymbol{\sigma}_m = (\sigma_{m1}, \ldots, \sigma_{mN}), \end{aligned} \tag{57}$$

i.e. the components are defined as products of univariate Gaussian densities

$$F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \prod_{n\in\mathcal{N}} f_n(x_n|\mu_{mn}, \sigma_{mn}), \tag{58}$$

$$f_n(x_n|\mu_{mn}, \sigma_{mn}) = \frac{1}{\sqrt{2\pi}\sigma_{mn}} \exp\left\{-\frac{(x_n - \mu_{mn})^2}{2(\sigma_{mn})^2}\right\}. \tag{59}$$

The corresponding weighted log-likelihood function (cf. (37)) is given by the equation

$$Q_{mn}(\mu_{mn}, \sigma_{mn}) = \sum_{x\in\mathcal{S}} \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} \log f_n(x_n|\mu_{mn}, \sigma_{mn}), \quad m \in \mathcal{M},\ n \in \mathcal{N}.$$

Again, the implicit equation (37) has a simple solution which can be derived as a weighted analogy of the well-known maximum-likelihood estimates (cf. Eq. (24)):

$$\mu'_{mn} = \sum_{x\in\mathcal{S}} x_n \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|}, \quad \left(w'_m|\mathcal{S}| = \sum_{x\in\mathcal{S}} q(m|\boldsymbol{x})\right), \tag{60}$$

$$(\sigma'_{mn})^2 = \sum_{x\in\mathcal{S}} (x_n - \mu'_{mn})^2 \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} = \sum_{x\in\mathcal{S}} x_n^2 \frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|} - (\mu'_{mn})^2. \tag{61}$$

As mentioned earlier, the Gaussian mixture model (57) is suitable for approximating unknown multimodal densities. The mixture parameters can be initialized with uniform weights and sufficiently large identical variances σ_{mn} to keep the data "visible" for all components. The components' means $\boldsymbol{\mu}_m$ can be chosen randomly as

a slight variation of the global means to avoid undesirable interference of the component parameter "trajectories" (cf. Sec. 3.3).

Recall that diagonal covariance matrices in Gaussian mixture components simplify the EM algorithm and decrease the risk of frequently occurring ill-conditioned matrices. In view of the obvious similarity to Parzen (kernel) estimates, the approximation capability of the conditional independence model (57) can be improved by increasing the number of components (cf. Sec. 1.1). We remark that the basic proof of asymptotic unbiasedness and consistency of the kernel estimates[56] also assumes the kernel function in a product form.

The diagonal covariance matrices could appear unsuitable when the Gaussian components should conform with narrow elongated "skew" clusters of data points. For this reason the variables are often normalized to zero means and unity variances. However, such a preprocessing of data is superfluous because the EM estimate of a Gaussian mixture density with the components (58) is invariant with respect to an arbitrary linear transform of variables (cf. Ref. 34, Appendix I).

The EM iteration equations (60) and (61) can be easily modified[12,24] to estimate the structural mixture model

$$P(\boldsymbol{x}|\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\phi}) = F(\boldsymbol{x}|\boldsymbol{\mu}_0,\boldsymbol{\sigma}_0)\sum_{m\in\mathcal{M}} w_m G(\boldsymbol{x}|\boldsymbol{\mu}_m,\boldsymbol{\sigma}_m,\boldsymbol{\phi}_m) \tag{62}$$

by introducing the "background" probability density in the form:

$$F(\boldsymbol{x}|\boldsymbol{\mu}_0,\boldsymbol{\sigma}_0) = \prod_{n\in\mathcal{N}} f_n(x_n|\mu_{0n},\sigma_{0n}), \tag{63}$$

$$\mu_{0n} = \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} x_n, \quad (\sigma_{0n})^2 = \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} (x_n-\mu_{0n})^2. \tag{64}$$

The related structural criterion (45) can be expressed (cf. (60) and (61)) in the form:

$$\gamma'_{mn} = \frac{w'_m}{2}\left[\frac{(\mu'_{mn}-\mu_{0n})^2+(\sigma'_{mn})^2}{(\sigma_{0n})^2} - 1 - 2\log\frac{\sigma'_{mn}}{\sigma_{0n}}\right] \tag{65}$$

which is equivalent to the following continuous version of the Kullback–Leibler information divergence:

$$\begin{aligned}\gamma'_{mn} &= w'_m\int_{\mathcal{X}_n} f_n(\xi|\mu'_{mn},\sigma'_{mn})\log\frac{f_n(\xi|\mu'_{mn},\sigma'_{mn})}{f_n(\xi|\mu_{0n},\sigma_{0n})}\,d\xi \\ &= w'_m I(f_n(\cdot|\mu'_{mn},\sigma'_{mn})||f_n(\cdot|\mu_{0n},\sigma_{0n})).\end{aligned} \tag{66}$$

We recall that the structural optimization criterion γ_{mn} in the general form (45) has been derived from the condition (47), which guarantees the monotonic property of EM algorithm.[16,32] From Eqs. (56) and (66) it follows that, in the case of both Bernoulli and Gaussian product mixtures, the structural criterion has a theoretical interpretation in terms of Kullback–Leibler information divergence. In this sense, at each iteration of EM algorithm only the most informative conditional distributions

$p'_n(\cdot|m)$ or density functions $f'_n(\xi\cdot|\mu_{mn},\sigma_{mn})$ are included in the components. The influence of the less relevant "noisy" variables is suppressed by introducing the corresponding terms of the common background.

4.4. *The EM algorithm for missing data*

The problem of missing data is a traditional domain of mathematical statistics, since most statistical methods cannot be applied to incomplete data vectors. The data can be made complete by simply omitting the incomplete vectors or variables but by doing this, we could lose a large part of the available information. Another possibility is to replace the missing values with some estimates.[5,70] However, as the estimated values have to be typical in a certain sense, the natural variability of data would be partly suppressed. In this situation, the product mixture model provides us with a simple possibility to apply the EM algorithm directly to incomplete data. It is not necessary to substitute for the missing values since the product components can be reduced to an arbitrary subspace currently available in $\boldsymbol{x}$. In other words, we estimate the mixture parameters using only the available data (cf. Ref. 28 for an example).

In order to modify the EM algorithm for incomplete data we denote by $\mathcal{N}(\boldsymbol{x})\subset\mathcal{N}$ the subset of indices of the defined variables in a given vector $\boldsymbol{x}$ and $\mathcal{S}_n\subset\mathcal{S}$ the subset of vectors with the defined value x_n:

$$\mathcal{N}(\boldsymbol{x})=\{n\in\mathcal{N}: x_n \text{ is defined in } \boldsymbol{x}\},\quad \mathcal{S}_n=\{\boldsymbol{x}\in\mathcal{S}: n\in\mathcal{N}(\boldsymbol{x})\},\quad n\in\mathcal{N}.$$

The modified EM iteration equations can be expressed in the form ($m\in\mathcal{M},\boldsymbol{x}\in\mathcal{S}$):

$$q(m|\boldsymbol{x})=\frac{w_m\prod_{n\in\mathcal{N}(x)}f_n(x_n|\theta_{mn})}{\sum_{j=1}^{M}w_j\prod_{n\in\mathcal{N}(x)}f_n(x_n|\theta_{jn})},\quad w'_m=\frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}q(m|\boldsymbol{x}),\tag{67}$$

$$Q_{mn}(\theta_{mn})=\sum_{x\in\mathcal{S}_n}\frac{q(m|\boldsymbol{x})}{w'_m|\mathcal{S}|}\log f_n(x_n|\theta_{mn}),\quad m\in\mathcal{M},\ n\in\mathcal{N},\tag{68}$$

$$\theta'_{mn}=\arg\max_{\theta_{mn}}\{Q_{mn}(\theta_{mn})\},\quad m\in\mathcal{M},\ n\in\mathcal{N}.\tag{69}$$

Roughly speaking, we calculate the values $q(m|\boldsymbol{x})$ in Eq. (67) only for the variables x_n currently available in $\boldsymbol{x}$ and the new parameters θ'_{mn} in Eq. (69) only from the data vectors $\boldsymbol{x}\in\mathcal{S}_n$ with the defined variable x_n. It can be seen that the proof of the monotonic property (cf. Sec. 2.1) is also applicable to Eqs. (67)–(69).

Obviously, there is a standard trade-off between the percentage of missing values and the estimation accuracy. The most suitable type of missing values is random with reasonable bounds:

$$N_{\min}=\min_{x\in\mathcal{S}}\{|\mathcal{N}(\boldsymbol{x})|\},\quad S_{\min}=\min_{n\in\mathcal{N}}\{|\mathcal{S}_n|\},$$

whereby the completely missing variables and "empty" vectors must be omitted.

The incomplete-data modification of EM algorithm has been used to compute the interactive statistical model of the Czech census data[28] containing about 1.5 million incomplete data records (questionnaires) with about 3 million missing answers (cf. Sec. 5.5).

5. Applications of Product Mixtures

The concept of structural product mixtures[12] has been proposed in the framework of statistical pattern recognition with the aim of Bayesian decision-making. Considering a statistical decision problem $\{P(\boldsymbol{x}|\omega)p(\omega), \omega \in \Omega\}$ with the *a priori* probabilities $p(\omega)$, we assume the class-conditional distributions $P(\boldsymbol{x}|\omega)$ in the form of structural product mixtures (cf. (38)–(40)):

$$P(\boldsymbol{x}|\omega) = P(\boldsymbol{x}|\omega, \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\phi}) = \sum_{m \in \mathcal{M}_\omega} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_0) G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m), \quad \omega \in \Omega, \quad \boldsymbol{x} \in \mathcal{X},$$

where $\mathcal{M}_\omega$ denotes the respective component index sets. The background probability density $F(\boldsymbol{x}|\boldsymbol{\theta}_0)$ can be reduced in the Bayes formula

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega, \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\phi})p(\omega)}{\sum_{\vartheta \in \Omega} P(\boldsymbol{x}|\vartheta, \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\phi})p(\vartheta)} = \frac{p(\omega) \sum_{m \in \mathcal{M}_\omega} w_m G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}{\sum_{\vartheta \in \Omega} p(\vartheta) \sum_{m \in \mathcal{M}_\vartheta} w_m G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}, \quad (70)$$

and therefore any decision-making may be confined to just the relevant variables. In other words, the Bayes decision function can be expressed as a weighted sum of component functions $G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)$ which can be defined on different subspaces:

$$\omega^* = d(\boldsymbol{x}) = \arg\max_{\omega \in \Omega} \{p(\omega|\boldsymbol{x})\} = \arg\max_{\omega \in \Omega} \left\{ p(\omega) \sum_{m \in \mathcal{M}_\omega} w_m G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) \right\}. \quad (71)$$

The primary motivation for the structural mixture model has been the problem of biologically unnatural complete interconnection of probabilistic neurons. We recall that all class-conditional probability distributions $P(\boldsymbol{x}|\omega)$ in the Bayes formula must be normed in the same space and therefore they must be defined in the same set of input variables.[32] To the best of our knowledge, the above mixture model is the only statistically correct way to remove this structural limitation. In the following subsections we illustrate the computational advantages of structural mixtures (cf. Sec. 4.1).

5.1. *Recognition of numerals in binary representation*

In recent years we have repeatedly applied multivariate Bernoulli mixtures for recognition of handwritten numerals from the NIST Special Database 19 (SD19) in order to verify different decision-making aspects.[25,26] The NIST benchmark database containing about 400,000 handwritten numerals in binary raster representation (about 40,000 for each numeral) has been normalized to a 32×32 binary raster to obtain 1024-dimensional binary data vectors. We have used the odd samples of each

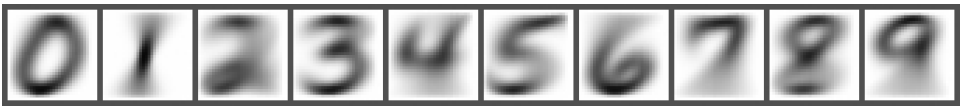

Fig. 1. Marginal probabilities of classes in raster arrangement ("mean images").

class for training and the even samples for testing to guarantee the same statistical properties of both datasets.[26] With the aim to increase the variability of the input binary patterns, we extended the training datasets four times by making three differently rotated variants of each pattern (by -4°, -2° and $+2^\circ$) with the resulting 80,000 training data vectors for each class. The marginal probabilities of classes in raster arrangement ("mean images") can be seen in Fig. 1.

For the sake of this paper we have solved the classification problem several times to document the limited influence of the randomly chosen initial values of parameters. In all experiments the initial number of components was set to $M = 200$ for each class and the component parameters were initialized randomly with uniform weights. In the first ten experiments I–X (cf. Table 1) the component parameters θ_{mn} were set with equal probability of either $\theta_{mn} = 0.51$ or $\theta_{mn} = 0.49$. In the experiments XI–XX the parameters were specified according to randomly chosen training numerals. For a given $\boldsymbol{x} \in \mathcal{S}$ we set: $\theta_{mn} = 0.51$ for $x_n = 1$ and $\theta_{mn} = 0.49$ for $x_n = 0$.

By using the randomly generated initial parameters we estimated in each experiment the ten class-conditional Bernoulli mixtures (in the subspace modification (39)) and then the accuracy of the resulting Bayes decision function (71) was characterized by the classification error matrix. Note that the number of components and the number of component-specific parameters may change in the course of EM iterations (cf. Secs. 4.1 and 4.2). The error matrix of Table 2 shows in the off-diagonal fields the mean values of false negative decisions along with the standard

Table 1. In all experiments I–XX the initial number of components was $M = 200$ and the mixture parameters were initialized randomly. *Components* denotes the resulting number of all components, *Parameters* denotes the resulting total number of mixture parameters in thousands and *Error* the resulting classification error in %. In the first ten experiments I–X the component parameters have been set with equal probability of either $\theta_{mn} = 0.51$ or $\theta_{mn} = 0.49$. In the experiments IX–XX the parameters were specified by randomly chosen training numerals: $x_n = 1 \Rightarrow \theta_{mn} = 0.51$; and $x_n = 0 \Rightarrow \theta_{mn} = 0.49$. The global mean error was 2.70%.

Experiment	I	II	III	IV	V	VI	VII	VIII	IX	X
Components	1645	1647	1653	1655	1664	1665	1672	1699	1758	1816
Parameters (in 10^3)	1460	1457	1464	1470	1469	1472	1476	1489	1491	1524
Error in %	**2.75**	**2.77**	**2.75**	**2.71**	**2.69**	**2.78**	**2.73**	**2.76**	**2.80**	**2.74**
Experiment	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX	XX
Components	1984	1985	1988	1994	1997	1998	1999	1999	2000	2000
Parameters (in 10^3)	1731	1735	1742	1760	1764	1765	1769	1768	1862	1862
Error in %	**2.67**	**2.74**	**2.72**	**2.73**	**2.60**	**2.67**	**2.71**	**2.62**	**2.57**	**2.61**

Table 2. Mean classification error matrix obtained in the randomly initialized estimation experiments I–XX from Table 1. The matrix illustrates the low variability of the resulting classifier properties. In each row the diagonal contains the mean number of correctly recognized numerals, the off-diagonal elements contain the mean numbers of the corresponding false negative decisions ± standard deviations. The last row contains the mean final numbers of components in the respective classes ± standard deviations.

C	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	**19,853**	10 ± 5	61 ± 13	19 ± 6	33 ± 6	43 ± 8	50 ± 10	4 ± 4	66 ± 13	36 ± 12
1	7 ± 3	**21,824**	45 ± 10	10 ± 4	70 ± 15	12 ± 7	59 ± 21	172 ± 42	82 ± 14	19 ± 7
2	36 ± 11	51 ± 12	**19,642**	71 ± 11	39 ± 8	10 ± 4	14 ± 6	30 ± 8	125 ± 11	24 ± 6
3	25 ± 7	22 ± 8	86 ± 8	**19,848**	4 ± 2	149 ± 9	4 ± 3	33 ± 7	288 ± 20	64 ± 6
4	38 ± 12	19 ± 12	19 ± 6	4 ± 3	**18,948**	7 ± 5	47 ± 10	74 ± 12	76 ± 9	358 ± 27
5	41 ± 11	26 ± 14	15 ± 6	206 ± 16	9 ± 3	**17,775**	53 ± 11	10 ± 6	174 ± 17	51 ± 9
6	98 ± 12	23 ± 10	24 ± 10	10 ± 6	54 ± 13	142 ± 16	**19,522**	2 ± 2	77 ± 9	6 ± 5
7	14 ± 5	20 ± 7	117 ± 10	27 ± 8	88 ± 8	3 ± 1	0 ± 1	**20,264**	41 ± 8	338 ± 26
8	40 ± 12	33 ± 19	56 ± 14	130 ± 10	21 ± 7	87 ± 11	19 ± 8	23 ± 12	**19,315**	70 ± 16
9	19 ± 6	20 ± 9	27 ± 5	96 ± 8	161 ± 16	34 ± 7	4 ± 2	200 ± 33	186 ± 13	**19,034**
M	181 ± 19	146 ± 49	197 ± 3	198 ± 2	192 ± 9	194 ± 7	179 ± 22	174 ± 27	196 ± 4	182 ± 20

deviations — as obtained in the 20 experiments. Despite the different randomly initialized parameters the classification results are very similar.

5.2. *Sequential recognition*

The sequential decision-making is a traditional problem of statistical pattern recognition.[7,2,63] Unlike in the standard classification scheme, the features are not considered all at once but only successively, one at a time. With the aim to reduce the number of feature measurements that are necessary for the final decision, we have to choose the most informative feature at any stage of classification. Consequently, the key problem of sequential recognition is the optimal online ordering of features, given a subset of available feature measurements. In the case of dependent variables the optimal choice of the most informative feature can be strongly influenced by the preceding measurements; therefore, it is of basic importance to know the underlying conditional probability distributions of the remaining features.

In statistical pattern recognition there are various methods to estimate the unknown class-conditional distributions or densities but, from the computational point of view, it is usually prohibitive to derive online the conditional marginals of the unobserved features, given a subset of previous feature measurements. Approximating the unknown class-conditional distributions by product mixtures we have a unique possibility to solve the online feature ordering problem by computing the individual Shannon informativity of the unobserved features for any given subset of preceding measurements.[21] In literature a comparable explicit solution of the online feature ordering problem is available only for the "naive Bayes" model[2] assuming the class-conditional independence of features. The general case of dependent features has been solved only approximately.[7,63]

In particular, assuming the class-conditional probability distribution $P(\boldsymbol{x}|\omega)$ in the form (35), a given subset of feature measurements

$$\boldsymbol{x}_C = (x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \in \mathcal{X}_C, \quad \mathcal{C} = \{i_1, \ldots, i_k\} \subset \mathcal{N}, \tag{72}$$

and an arbitrary unobserved variable $x_n (n \in \mathcal{N} \backslash \mathcal{C})$, we can write[21] the explicit formulae for the conditional distributions $P_{n|C}(x_n|\boldsymbol{x}_C), P_{n|C\omega}(x_n|\boldsymbol{x}_C, \omega)$ as

$$P_{n|C}(x_n|\boldsymbol{x}_C) = \frac{P_{n,C}(x_n, \boldsymbol{x}_C)}{P_C(\boldsymbol{x}_C)} = \sum_{\omega \in \Omega} \sum_{m \in \mathcal{M}_\omega} \bar{W}_m^{\omega}(\boldsymbol{x}_C) f_n(x_n|m), \tag{73}$$

$$P_{n|C\omega}(x_n|\boldsymbol{x}_C, \omega) = \frac{P_{n,C|\omega}(x_n, \boldsymbol{x}_C|\omega)}{P_{C|\omega}(\boldsymbol{x}_C|\omega)} = \sum_{m \in \mathcal{M}_\omega} W_m^{\omega}(\boldsymbol{x}_C) f_n(x_n|m), \tag{74}$$

where

$$W_m^{\omega}(\boldsymbol{x}_C) = \frac{w_m F_C(\boldsymbol{x}_C|m)}{\sum_{j \in \mathcal{M}_\omega} w_j F_C(\boldsymbol{x}_C|j)}, \quad \bar{W}_m^{\omega}(\boldsymbol{x}_C) = \frac{p(\omega) w_m F_C(\boldsymbol{x}_C|m)}{\sum_{\vartheta \in \Omega} p(\vartheta) \sum_{j \in \mathcal{M}_\vartheta} w_j F_C(\boldsymbol{x}_C|j)}.$$

By using Eqs. (73) and (74) we can write the Shannon formula for the conditional information $I_{\boldsymbol{x}_C}(\mathcal{X}_n, \Omega)$ about Ω contained in the variable x_n given the subvector $\boldsymbol{x}_C \in \mathcal{X}_C$:

$$I_{\boldsymbol{x}_C}(\mathcal{X}_n, \Omega) = H_{\boldsymbol{x}_C}(\Omega) - H_{\boldsymbol{x}_C}(\Omega|\mathcal{X}_n) = H_{\boldsymbol{x}_C}(\mathcal{X}_n) - H_{\boldsymbol{x}_C}(\mathcal{X}_n|\Omega). \tag{75}$$

Here $H_{\boldsymbol{x}_C}(\mathcal{X}_n)$ and $H_{\boldsymbol{x}_C}(\mathcal{X}_n|\Omega)$ are the respective Shannon entropies:

$$H_{\boldsymbol{x}_C}(\mathcal{X}_n) = \sum_{x_n \in \mathcal{X}_n} -P_{n|C}(x_n|\boldsymbol{x}_C) \log P_{n|C}(x_n|\boldsymbol{x}_C), \tag{76}$$

$$H_{\boldsymbol{x}_C}(\mathcal{X}_n|\Omega) = \sum_{\omega \in \Omega} P_{\Omega|C}(\omega|\boldsymbol{x}_C) \sum_{x_n \in \mathcal{X}_n} -P_{n|C\omega}(x_n|\boldsymbol{x}_C, \omega) \log P_{n|C\omega}(x_n|\boldsymbol{x}_C, \omega). \tag{77}$$

Finally we can choose the next most informative feature measurement x_{n_0}:

$$n_0 = \arg \max_{n \in \mathcal{N} \backslash \mathcal{C}} \{I_{\boldsymbol{x}_C}(\mathcal{X}_n, \Omega). \tag{78}$$

which guarantees the maximum expected decrease of uncertainty of the posterior probability distribution $p(\omega|\boldsymbol{x}_C, x_n)$ expressed by the entropy $H_{\boldsymbol{x}_C}(\Omega|\mathcal{X}_n)$ (cf. (75)).

On the base of the solution from Sec. 5.1, Fig. 2 illustrates the proposed method of sequential recognition of numerals on the binary 32×32-raster. Note that the expected image of the numeral two may strongly change in the case of an unexpected value of the uncovered raster field.

One of the most natural application domains of sequential pattern recognition is medical diagnostics. The computer-aided medical decision-making usually includes a large number of possible diagnoses and diagnostic features, and therefore the main purpose of a sequential procedure is to choose the maximum possible diagnostic information. In this respect the information-controlled sequential questioning is capable of efficiently finding the diagnostically relevant features. The resulting

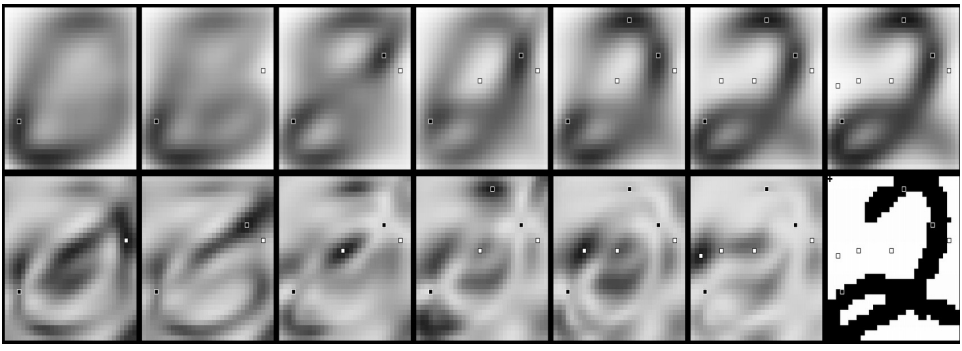

Fig. 2. Sequential recognition of the numeral two. Note that only seven raster fields are sufficient to recognize the numeral. The first row shows the changing expectation of the classifier. The second row shows the informativity of raster fields corresponding to the currently uncovered (white or black) raster fields and finally the input image.

posterior distribution $p(\omega|\boldsymbol{x}_C)$ can be used as input information of a medical expert for the sake of a final diagnosis.[22]

5.3. *Probabilistic neural networks*

Within the framework of statistical pattern recognition, we have developed the concept of probabilistic neural networks in a series of papers[15,16,19,25–27,30,32] as one of the most appealing applications of the structural mixture model. In particular, in a statistical decision problem $\{P(\boldsymbol{x}|\omega)p(\omega), \omega \in \Omega\}$, we assume the class-conditional structural mixtures in the form (cf. (70)):

$$P(\boldsymbol{x}|\omega) = P(\boldsymbol{x}|\omega, \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\phi}) = \sum_{m \in \mathcal{M}_\omega} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_0) G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m), \quad \omega \in \Omega, \ \boldsymbol{x} \in \mathcal{X}.$$

As the background probability density $F(\boldsymbol{x}|\boldsymbol{\theta}_0)$ can be reduced in the Bayes formula (70), we can express the Bayes decision function as a weighted sum of component functions defined on different subspaces (cf. (71)). Thus, within the framework of multilayer neural networks, the input connections of neurons can be confined to arbitrary subsets of input neurons whereby the structural optimization of neural networks can be included in the EM algorithm in full generality (cf. Sec. 4.1).

The basic principle of PNN is the interpretation of product components in terms of functional properties of biological neurons. We have shown that the estimated product mixtures (39) can be used to define an information-preserving transform in terms of *a posteriori* probabilities $q(m|\boldsymbol{x})$ (cf. (42)):

$$\boldsymbol{T} : \mathcal{X} \to \mathcal{Y}, \quad \mathcal{Y} \subset R^M, \quad \mathbf{y} = \boldsymbol{T}(\boldsymbol{x}) = (T_1(\boldsymbol{x}), T_2(\boldsymbol{x}), \ldots, T_M(\boldsymbol{x})) \in \mathcal{Y}, \quad (79)$$

$$y_m = T_m(\boldsymbol{x}) = \log q(m|\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X}, \ m \in \mathcal{M}. \quad (80)$$

The vector transform $\boldsymbol{T}$ maps the space $\mathcal{X}$ of neural inputs into the space of neural outputs $\mathcal{Y}$ and can be constructed repeatedly to design multilayer PNN.

Roughly speaking, the transform in (79) and (80) unifies the points $\boldsymbol{x} \in \mathcal{X}$ with identical posterior distributions $q(m|\boldsymbol{x})$ and therefore the decision information with respect to both components $I(\mathcal{X}, \mathcal{M})$ and the classes $I(\mathcal{X}, \Omega)$ is preserved[19,67]:

$$I(\mathcal{X}, \mathcal{M}) = I(\mathcal{Y}, \mathcal{M}), \quad I(\mathcal{X}, \Omega) = I(\mathcal{Y}, \Omega). \tag{81}$$

Simultaneously the entropy $H(\mathcal{Y})$ of the output space is minimized. In view of the structural mixture model described in Sec. 4.1 the incomplete interconnection structure of the transformation $\boldsymbol{T}$ can be optimized in a statistically correct way[16,32] by maximizing the log-likelihood criterion. Moreover, the transform (80) is fault-tolerant in the sense that small inaccuracies of the components may cause only bounded information loss. In particular, if the estimated parameters $\tilde{w}_m \tilde{q}(m|\boldsymbol{x})$ satisfy the condition

$$|T_m(\boldsymbol{x}) - \ln(\tilde{q}(m|\boldsymbol{x}) + \tilde{w}_m \delta)| < \epsilon, \quad m \in \mathcal{M}, \quad \boldsymbol{x} \in \mathcal{X}, \tag{82}$$

for some $\delta > 0$ and $\epsilon > 0$, then the resulting information loss is bounded by the inequality

$$I(\mathcal{X}, \mathcal{M}) - I(\mathcal{Y}, \mathcal{M}) < \delta + 2\epsilon. \tag{83}$$

From the point of view of neurophysiological interpretation, the posterior probability $q(m|\boldsymbol{x})$ is a natural measure of excitation of the mth neuron, given the vector of input stimuli $\boldsymbol{x} \in \mathcal{X}$. Thus, in view of Eq. (80), the output signal y_m of the mth neuron is defined as a logarithm of its excitation and we can write

$$y_m = \log w_m + \sum_{n \in \mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|\theta_{mn})}{f_n(x_n|\theta_{0n})} - \log \left(\sum_{j \in \mathcal{M}} G(\boldsymbol{x}|\boldsymbol{\theta}_j, \phi_j) w_j \right). \tag{84}$$

The first term in the last formula may correspond to spontaneous activity of the neuron and the second term sums up the afferent contributions of the neural inputs x_n. Here the nonzero structural parameters $\phi_{mn} = 1$ define the set of input variables, i.e. the receptive field of the mth neuron. Consequently, the term

$$g_{mn}(x_n) = \log \frac{f_n(x_n|\theta_{mn})}{f_n(x_n|\theta_{0n})} = \log f_n(x_n|\theta_{mn}) - \log f_n(x_n|\theta_{0n}) \tag{85}$$

represents the synaptic weight of the contribution x_n on the input of the mth neuron. The effectiveness of the synaptic transmission $g_{mn}(x_n)$ combines the statistical properties of the stimulus x_n with the activity of the "postsynaptic" neuron m. It is high when the input signal x_n frequently "supports" the excitation of the mth neuron, and it is low when x_n does not take part in its excitation. Finally, the last norming term in Eq. (84) corresponds to SC lateral inhibition and is responsible for the competitive properties of neurons. In this sense, Eq. (84) can be viewed as a statistical justification of the following classical Hebb's postulate of learning[45]:

> *When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic*

changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.

In a series of papers we have shown that: independently trained PNN can be combined both horizontally and vertically,[30] weighting of data is compatible with PNN[33] in a way allowing for selective evaluation of training data ("emotional" learning) and the design of PNN can be explained in terms of strictly modular sequential learning,[29] as discussed in the next subsection. An important "neuromorphic" feature of PNN is the invariance of structural mixtures with respect to the arbitrary permutation of variables.[30] Recall that in biological neural networks the information about the topological arrangement of input-layer neurons is not available at higher levels.

5.4. *Learning by sequential version of EM algorithm*

The concept of machine learning can be traced back to the early neural network models like Rosenblatt's perceptron — with a clearly defined sequential adaptation of input neural weights. In recent decades the term "learning" has become vastly inflated because essentially any data-oriented complex algorithm can be repeatedly applied to an extending dataset and therefore can be viewed as an instance of machine learning.

In connection with the probabilistic neural networks we refer to modular learning[17,19,29] to emphasize that, assuming an infinite sequence of data: (a) the learning is decentralized and sequential, without any necessity of storing input data; (b) the probabilistic neuron may adapt only its internal parameters; and (c) the adaptation information must be available at the neurons' interior or at its inputs.

The modular learning of probabilistic neurons is based on a sequential modification of EM algorithm which can be viewed as a single "infinite" iteration with intermediate updating of parameters. In its original form, the EM algorithm is a typical offline procedure repeatedly using all data in each iteration. However, assuming an infinite sequence of data

$$\{\boldsymbol{x}^{(t)}\}_{t=0}^{\infty}, \quad \boldsymbol{x}^{(t)} = (x_1, x_2, \ldots, x_N) \in \mathcal{X},\ t = 0, 1, 2, \ldots,$$

we can write ($m \in \mathcal{M}$):

$$q(m|\boldsymbol{x}^{(t)}) = \frac{\tilde{w}_m F(\boldsymbol{x}^{(t)}|\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\sigma}}_m)}{\sum_{j\in\mathcal{M}} \tilde{w}_j F(\boldsymbol{x}^{(t)}|\tilde{\boldsymbol{\mu}}_j, \tilde{\boldsymbol{\sigma}}_j)}, \quad \beta_m(\boldsymbol{x}^{(t)}) = \frac{q(m|\boldsymbol{x}^{(t)})}{\sum_{i=1}^{t} q(m|\boldsymbol{x}^{(i)})}, \tag{86}$$

where $\tilde{w}_m, \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\sigma}}_m$ are the current component parameters and the EM iteration equations can be rewritten in the following sequential form ($t = 1, 2, \ldots$):

$$w_m^{(t)} = [1 - \alpha(t)] w_m^{(t-1)} + \alpha(t) q(m|\boldsymbol{x}^{(t)}), \quad \alpha(t) = \frac{1}{t}, \tag{87}$$

$$\boldsymbol{\mu}_m^{(t)} = [1 - \beta_m(\boldsymbol{x}^{(t)})] \boldsymbol{\mu}_m^{(t-1)} + \beta_m(\boldsymbol{x}^{(t)}) \boldsymbol{x}^{(t)}, \quad m \in \mathcal{M}, \tag{88}$$

$$\boldsymbol{\sigma}_m^{(t)} = [1 - \beta_m(\boldsymbol{x}^{(t)})] \boldsymbol{\sigma}_m^{(t-1)} + \beta_m(\boldsymbol{x}^{(t)}) (\boldsymbol{x}^{(t)} - \tilde{\boldsymbol{\mu}}_m)^2, \tag{89}$$

where the sequentially computed parameters $w_m^{(t)}, \boldsymbol{\mu}_m^{(t)}, \boldsymbol{\sigma}_m^{(t)}$ are substituted periodically at some instants T_j:

$$\tilde{w}'_m = w_m^{(T_j)}, \quad \tilde{\boldsymbol{\mu}}'_m = \boldsymbol{\mu}_m^{(T_j)}, \quad \tilde{\boldsymbol{\sigma}}'_m = \boldsymbol{\sigma}_m^{(T_j)}, \quad j = 1, 2, 3, \ldots . \tag{90}$$

It is easily verified that, for a periodically repeating set of K data vectors $\boldsymbol{y} \in \mathcal{S}$:

$$\{\boldsymbol{x}^{(t)}\}_{t=0}^{\infty}, \quad \boldsymbol{x}^{(t)} = \boldsymbol{y}^{(k)} \in \mathcal{S},\ k = (t \bmod K) + 1,\ t = 0, 1, 2, \ldots , \tag{91}$$

and periodical substitution instants $T_i = iK$, the scheme (86)–(90) is equivalent to the standard EM algorithm if we replace the index t by the respective index k, and the vector $\boldsymbol{x}^{(t)}$ by the corresponding $\boldsymbol{y}^{(k)}$. Note that for $k = 1$ we obtain $\beta_m(\boldsymbol{y}^{(1)}) = \alpha(1) = 1$, i.e. the corresponding initial values $w_m^{(0)}, \boldsymbol{\mu}_m^{(0)}, \boldsymbol{\sigma}_m^{(0)}$ are multiplied by zero and therefore irrelevant.

Let us remark that the above sequential scheme is applicable to the general EM algorithm, but it is data-dependent and therefore the basic monotonic property of EM iterations is not guaranteed. However, in some cases, the sequential scheme yields even better results than the standard EM procedure because the underlying sequential computation includes more data than the standard EM algorithm which is confined to a given finite training set (cf. Ref. 29).

Generally, in the case of a long data sequence the substitution intervals $(T_{i+1}$–$T_i)$ should increase in the course of iterations to suppress random fluctuations. In our numerical experiments we have observed that the substitution intervals increasing with a coefficient chosen from the interval $\langle 1.2, 1.4\rangle$ yield good results without essential differences.[29]

The above periodical substitution principle has a plausible neurophysiological interpretation because, according to the Hebb's postulate of learning, the neural adaptation follows from *some growth process or metabolic changes* and therefore a delay can be expected between the primary activity of a neuron and the corresponding adaptive changes.

5.5. *Probabilistic expert systems*

Recently the product mixture model was applied to reproduce the results of the Czech census.[28] The source database containing 10,230,060 vectors (questionnaires) of 24 categorical variables (questions) has been used to estimate the discrete product mixture distribution. The estimated mixture is directly applicable as a knowledge base of the probabilistic expert system (PES)[13,14] and, by using the probabilistic inference mechanism, we can derive the statistical properties of various subpopulations in terms of conditional probability distributions (conditional histograms). In particular, given a subpopulation specified by the values of several variables (answers) $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$:

$$\boldsymbol{x}_C = (x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \in \mathcal{X}_C, \quad \mathcal{C} = \{i_1, \ldots, i_k\} \subset \mathcal{N}, \tag{92}$$

we can compute the conditional distributions $P_{n|C}(x_n|\boldsymbol{x}_C)$ for any of the remaining variables $x_n (n \notin \mathcal{C})$ by Eq. (73). The discrete conditional distributions can be displayed as histograms (one- or two-dimensional) playing the role of basic communication means. The estimated product mixture does not contain the original protected data and therefore the interactive user software can be distributed without any confidentiality concerns.

The original census database included 1,524,240 incomplete data records (questionnaires) with about 3 million missing values. For this reason we have used the "missing data" modification of EM algorithm (cf. Sec. 4.4) to estimate a large mixture model of $M = 10\,000$ components. We recall that there is no risk of "overfitting" if we try to reproduce the statistical properties. The main issue of the paper[28] was the model accuracy. By comparing the empirical frequencies with the estimated model probabilities we obtained the mean relative error of 4% which corresponds to the accuracy of any displayed histogram column. The resulting interactive model is available at the website http://ro.utia.cas.cz/census/.

The purpose of the proposed interactive method is to reproduce the statistical properties of arbitrary subpopulations without any risk of confidentiality violation. However, the estimated product mixture provides additional tools to analyze the data. Thus, the system automatically creates a virtual list including a large number of subpopulations which can be ordered according to different criteria like conditional probability of a value, conditional entropy of a variable or mutual informativity of a pair of variables. In this sense, qualitatively new possibilities arise to identify causal relations between values and variables. The proposed tools of information analysis are actually enabled by the statistical model of data in the form of a product mixture since any comparable evaluation of the original data would be exceedingly time-consuming.

5.6. *Image processing*

A feature common to both image processing and application of product mixtures to textures is the estimation of local properties. The concept of texture intuitively suggests some local shift-invariant statistical properties and simultaneously a global homogeneity in a certain sense. Motivated by this idea we describe the local statistical dependencies between pixels in terms of joint probability density of grey-levels in a suitably chosen (square or roughly square) shifting window. The unknown probability density can be estimated in the form of a multivariate Gaussian product mixture from a large set of data vectors obtained by shifting the window within the original image. Recall that, by using the EM algorithm, we implicitly assume independent data but, in the case of a shifting window, this assumption is violated because the neighboring windows overlap. This theoretical detail has specific consequences,[24,41] especially for texture modeling.

5.6.1. *Texture synthesis*

More specifically, let $\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{R}^N$ be a vector of grey-levels of the window in a specified order. Having estimated the Gaussian mixture model (57):

$$P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m), \quad \boldsymbol{x} \in \mathcal{R}^N, \tag{93}$$

we can synthesize artificial textures by sequential prediction. In particular, let $\boldsymbol{x}_A = (x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ be a given part of texture in the window then the remaining empty part $\boldsymbol{x}_B = (x_{j_1}, x_{j_2}, \ldots, x_{j_r})$ can be estimated by means of the conditional expectation formula. We can compute the conditional density

$$P_{B|A}(\boldsymbol{x}_B|\boldsymbol{x}_A) = \frac{P_{B,A}(\boldsymbol{x}_B, \boldsymbol{x}_A)}{P_A(\boldsymbol{x}_A)} = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_A) F_B(\boldsymbol{x}_B|\boldsymbol{\mu}_{mB}, \boldsymbol{\sigma}_{mB}), \tag{94}$$

$$W_m(\boldsymbol{x}_A) = \frac{w_m F_A(\boldsymbol{x}_A|\boldsymbol{\mu}_{mA}, \boldsymbol{\sigma}_{mA})}{\sum_{j \in \mathcal{M}} w_j F_A(\boldsymbol{x}_A|\boldsymbol{\mu}_{jA}, \boldsymbol{\sigma}_{jA})}, \quad \boldsymbol{\mu}_{mA} = (\mu_{m,i_1}, \ldots, \mu_{m,i_k}), \tag{95}$$

$$F_B(\boldsymbol{x}_B|\boldsymbol{\mu}_{mB}, \boldsymbol{\sigma}_{mB}) = \prod_{n \in \mathcal{N}_B} f_n(x_n|\mu_{mn}, \sigma_{mn}), \quad \boldsymbol{\mu}_{mB} = (\mu_{m,j_1}, \ldots, \mu_{m,j_r}),$$

and the unknown part of the window $\boldsymbol{x}_B$ can be estimated as a conditional expectation

$$E\{\boldsymbol{x}_B\} = \int_{\mathcal{R}^N} \boldsymbol{x}_B P_{B|A}(\boldsymbol{x}_B|\boldsymbol{x}_A) d\boldsymbol{x}_B = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_A) \boldsymbol{\mu}_{mB}. \tag{96}$$

Thus, the estimated part of the window $E\{\boldsymbol{x}_B\}$ can be expressed as a weighted sum of the corresponding parts of the component means $\boldsymbol{\mu}_{mB}$. In numerical experiments the resulting dimension is usually large and therefore the weighted sum (96) can be reduced to a single term in view of the nearly binary properties of the component weights $W_m(\boldsymbol{x}_A)$. In this way, starting with a single piece of texture (a seed), we can synthesize arbitrarily large textures by prediction, e.g. by shifting the square window stepwise left-to-right and top-to-down (cf. Fig. 3). The resulting texture is "smooth" because the component means $\boldsymbol{\mu}_m$ correspond to weighted sums of window "patches" (cf. (8)). In order to recover the high-frequency details of the original texture we replaced the component means $\boldsymbol{\mu}_m$ by the most similar patches from the original texture image. It can be seen that the resulting image looks more realistic (cf. Fig. 3, right lower part). Analogously we can synthesize color textures or even BTF textures.[41]

Obviously, the principle of local prediction can be applied analogously to any missing part of the image. In case of this type of problems called image inpainting we only need to specify, at any position of the window, the missing part to be estimated. Figure 4 illustrates the inpainting of a color picture containing several different types of textures. We have used a window of 13×13 pixels with trimmed corners

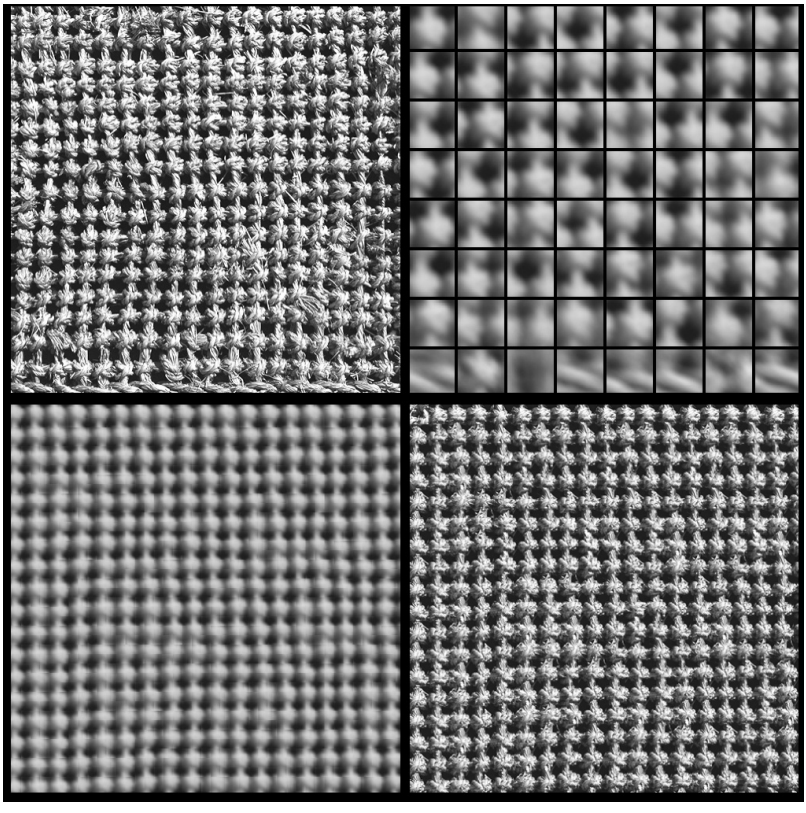

Fig. 3. Synthesis of the texture "ratan." In the upper part is the original texture image (left, 512×512 pixels) and the typical component means of the estimated mixture (right) in window arrangement (window size of 30×30 pixels, dimension $N = 900$ and number of components $M = 87$). In the lower part is the "smooth" synthesized texture (left) and the "realistic" texture model based on the component-related "centroids" (right).

containing 145 pixels in three spectral values. It can be seen that the resulting 435-dimensional mixture model having 223 product components automatically identifies the different types of textures.[40]

5.6.2. *Log-likelihood evaluation of images*

The texture synthesis by local prediction provides a unique possibility to verify the quality of the underlying statistical model visually. Comparing the original and synthesized texture in numerous experiments we have found that the Gaussian product mixtures are capable of describing the local image properties to a high degree of accuracy. Motivated by the good experimental results we have proposed to apply the estimated mixture model to the original data, our goal being to evaluate the "typicality" of different locations in the original image. In particular, at each position of the shifting window we compute the estimated density $P(\boldsymbol{x})$, which can be viewed

Fig. 4. Example of image inpainting. In the upper part is the final "inpainted" image (1280 × 960 pixels, model dimension $N = 435$ and $M = 223$ components) and in the lower part is the damaged source image containing some missing parts.

as a measure of typicality of the window interior, and the log-likelihood value $\log P(\boldsymbol{x})$ is displayed at the central pixel of the window in suitable scaling. The resulting log-likelihood image is statistically well justified and easily interpretable since the dark places correspond to "unusual" or "atypical" parts and light pixels reflect the more typical or more probable locations. The log-likelihood mapping has been applied to the evaluation of screening mammograms[20,34,44] (cf. Fig. 5) and to image forgery detection.[39] The local statistical model is also applicable to segmentation of images.[37]

Alternatively we could display the log-likelihood ratio values $\log P(\boldsymbol{x})/P_0(\boldsymbol{x})$ where $P_0(\boldsymbol{x})$ is the product of univariate global marginals. The log-likelihood ratio image avoids a direct dependence on grey-levels and is more sensitive to structural irregularities.[36] The function $\log P(\boldsymbol{x})/P_0(\boldsymbol{x})$ can also be used to identify different properties in one-class classifiers.[26]

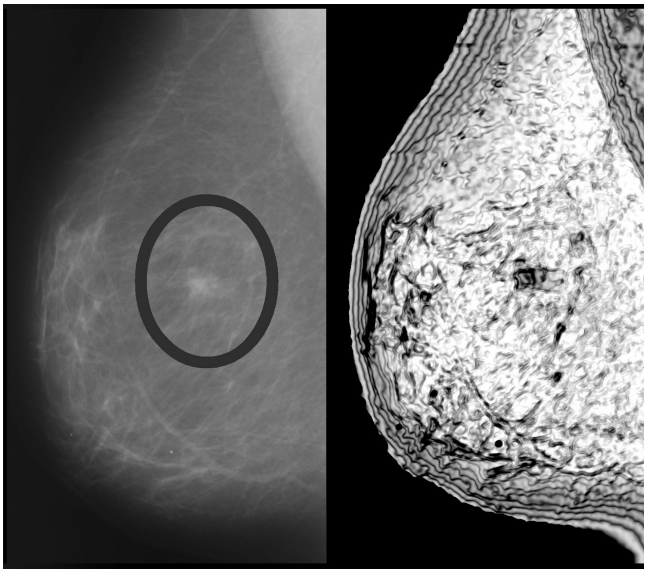

Fig. 5. Original image of the mammogram C-0143-1 (left) from the DDSM database. In the original mammogram there is a circumscribed malignant mass. In the log-likelihood image (right) the pixel values are defined by the respective log-likelihood values $\log P(\mathbf{x})$ where $\mathbf{x}$ is the 145-dimensional vector defined by the shifting search window. The malignant lesion is partly emphasized by contour lines resulting as a side-effect from the pixel-wise evaluation of the mammogram.

6. Concluding Remarks

The estimation of multivariate distribution mixtures in spaces of high dimensionality ($N \approx 10^2$) is a difficult task, from both the theoretical and computational points of view. Even if we succeed in managing the computational instability of EM algorithm (as discussed in Sec. 3.4), we are still confronted with the specific properties of the estimated mixtures in high-dimensional spaces.

As the components usually correspond to small separated subsets of data points in the sparse high-dimensional spaces, the "generalizing properties" of high-dimensional mixtures are often poor. We have obtained good results in recognition of handwritten numerals on a binary raster with large training datasets,[25,26,32] but the synthesis of artificial textures based on a local statistical model often fails.[24,41] We recall that, in the case of texture synthesis, the training dataset is obtained by shifting a square window through the source texture image, and therefore the data vectors defined by overlapping windows are not independent. Actually, the training data in the log-likelihood function correspond to a specific "trajectory" in the sample space produced by the shifting window. As the data points generated by prediction may happen to be "atypical," the corresponding component values are often very low and the prediction becomes unreliable. For this reason we can achieve more successful applications when the estimated high-dimensional mixture model is applied to

the original data, e.g. with the goal of reproducing the statistical properties of databases[28] or to evaluate local statistical properties of a given image.[20,34,36,39]

As noted in Sec. 3.3, the EM iterations can be stopped by thresholding the relative increment of the log-likelihood criterion without any substantial loss of approximation accuracy. Nevertheless, the estimation of high-dimensional mixtures with many components from large datasets may become time-consuming. In this respect we remark that the EM algorithm can be easily parallelized by a suitable decomposition of the available dataset.

Acknowledgments

This work was supported by the Czech Science Foundation Project No. 17-18407S.

References

1. S. A. Ajvazjan, Z. I. Bezhaeva and O. V. Staroverov, *Classification of Multivariate Observations*, (Statistika, Moscow, 1974) (in Russian).
2. M. Ben-Bassat, Pattern-based interactive diagnosis of multiple disorders: The MEDAS system, *IEEE Trans. Pattern Anal. Mach. Intell.* **2**(2) (1980) 148–160.
3. N. Bouguila, D. Ziou and J. Vaillancourt, Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application, *IEEE Trans. Image Process.* **13**(11) (2004) 1533–1543.
4. N. E. Day, Estimating the components of a mixture of normal distributions, *Biometrika* **56** (1969) 463–474.
5. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc. B* **39** (1977) l–38.
6. M. A. T. Figueiredo and A. K. Jain, Unsupervised learning of finite mixture models, *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(3) (2002) 381–396.
7. K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning* (Academic Press, New York, 1968).
8. N. Gallego-Ortiz and D. S. Femandez-Mc-Cann, Efficient implementation of the EM algorithm for mammographic image texture analysis with multivariate Gaussian mixtures, in *Proc. 2011 IEEE Statistical Signal Processing Workshop (SSP)* (2011), pp. 821–824.
9. M. W. Graham and D. J. Miller, Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection, *IEEE Trans. Signal Process.* **54** (4) (2006) 1289–1303.
10. J. Grim, On numerical evaluation of maximum-likelihood estimates for finite mixtures of distributions, *Kybernetika* **l8**(3) (1982) 173–190, http://dml.cz/dmlcz/124132.
11. J. Grim, On structural approximating multivariate discrete probability distributions, *Kybernetika* **20**(1) (1984) 1–17, http://dml.cz/dmlcz/125676.
12. J. Grim, Multivariate statistical pattern recognition with nonreduced dimensionality, *Kybernetika* **22**(2) (1986) 142–157, http://dml.cz/dmlcz/125022.
13. J. Grim, A dialog presentation of census results by means of the probabilistic expert system PES, in *Proc. Eleventh European Meeting on Cybernetics and Systems Research*, ed. R. Trappl (World Scientific, Singapore, 1992), pp. 997–1005.
14. J. Grim, Knowledge representation and uncertainty processing in the probabilistic expert system PES, *Int. J. Gen. Syst.* **22**(2) (1994) 103–111.

15. J. Grim, Design of multilayer neural networks by information preserving transforms, in *Proc. Third European Congr. Systems Science*, eds. E. Pessa, M. P. Penna and A. Montesanto (Edizioni Kappa, Roma, 1996) pp. 977–982.
16. J. Grim, Information approach to structural optimization of probabilistic neural networks, in *Proc. 4th System Science European Congr.*, eds. L. Ferrer *et al.* (Sociedad Espanola de Sistemas Generales, Valencia, 1999) pp. 527–540.
17. J. Grim, A sequential modification of EM algorithm, in *Studies in Classification, Data Analysis and Knowledge Organization*, eds. W. Gaul and H. Locarek-Junge (Springer, 1999), pp. 163–170.
18. J. Grim, EM cluster analysis for categorical data, in *Structural, Syntactic and Statistical Pattern Recognition.*, eds. D. Y. Yeung, J. T. Kwok and A. Fred, LNCS, Vol. 4109 (Springer, Berlin, 2006), pp. 640–648.
19. J. Grim, Neuromorphic features of probabilistic neural networks, *Kybernetika* **43**(5) (2007) 697–712, http://dml.cz/dmlcz/135807.
20. J. Grim, Preprocessing of screening mammograms based on local statistical models, *Proc. 4th Int. Symp. Applied Sciences in Biomedical and Communication Technologies (ISABEL 2011)* (ACM, Barcelona, 2011), pp. 1–5.
21. J. Grim, Sequential pattern recognition by maximum conditional informativity, *Pattern Recognit. Lett.* **45C** (2014) 39–45, http://library.utia.cas.cz/separaty/2014/RO/grim-0428565.pdf.
22. J. Grim, Feasibility study of an interactive medical diagnostic Wikipedia, in *Stochastic and Physical Monitoring Systems (SPSM 2016)* (CTU, Prague, 2016), pp. 31–45.
23. J. Grim, Distribution mixtures I and II (2014), http://www.utia.cas.cz/people/grim.
24. J. Grim, M. Haindl, P. Somol and P. Pudil, A subspace approach to texture modelling by using Gaussian mixtures, in *Proc. 18th IAPR Int. Conf. Pattern Recognition (ICPR 2006)*, eds. B. Haralick and T. K. Ho (IEEE Computer Society, Los Alamitos, 2006), pp. 235–238.
25. J. Grim and J. Hora, Iterative principles of recognition in probabilistic neural networks, *Neural Netw.* **21**(6) (2008) 838–846.
26. J. Grim and J. Hora, Recognition of properties by probabilistic neural networks, in *Artificial Neural Networks: ICANN 2009*, LNCS, Vol. 5769 (Springer, 2009), pp. 165–174.
27. J. Grim and J. Hora, Computational properties of probabilistic neural networks, in *Artificial Neural Networks: ICANN 2010*, LNCS, Vol. 5164 (Springer, Berlin, 2010), pp. 52–61.
28. J. Grim, J. Hora, P. Boček, P. Somol and P. Pudil, Statistical model of the 2001 Czech census for interactive presentation, *J. Off. Stat.* **26**(4) (2010) 673-694.
29. J. Grim, P. Just and P. Pudil, Strictly modular probabilistic neural networks for pattern recognition, *Neural Netw. World* **13**(6) (2003) 599–616.
30. J. Grim, J. Kittler, P. Pudil and P. Somol, Multiple classifier fusion in probabilistic neural networks, *Pattern Anal. Appl.* **5**(7) (2002) 221–233.
31. J. Grim and P. Pudil, Mixtures of product components versus mixtures of dependence trees, in *Computational Intelligence* (Springer, Cham, 2015), pp. 365–382.
32. J. Grim, P. Pudil and P. Somol, Recognition of handwritten numerals by structural probabilistic neural networks, in *Proc. 2nd ICSC Symp. Neural Computation*, eds. H. Bothe and R. Rojas (ICSC, Wetaskiwin, 2000), pp. 528–534.
33. J. Grim, P. Pudil and P. Somol, Boosting in probabilistic neural networks, in *Proc. 16th Int. Conf. Pattern Recognition*, eds. R. Kasturi, D. Laurendeau and C. Suen (IEEE Computer Society, Los Alamitos, 2002), pp. 136–139.

34. J. Grim, P. Somol, M. Haindl and J. Daneš, Computer-aided evaluation of screening mammograms based on local texture models, *IEEE Trans. Image Process.* **18**(4) (2009) 765–773.
35. J. Grim, P. Somol, M. Haindl and J. Daneš, Evaluation of mammograms (demo), http://ro.utia.cas.cz/.
36. J. Grim, P. Somol, M. Haindl and P. Pudil, A statistical approach to local evaluation of a single texture image, in *Proc. 16th Annu. Symp. PRASA 2005*, ed. F. Nicolls (University of Cape Town, 2005), pp. 171–176, http://www.prasa.uct.ac.za/.
37. J. Grim, P. Somol, M. Haindl and P. Pudil, Color texture segmentation by decomposition of Gaussian mixture model, in *Progress in Pattern Recognition, Image Analysis and Applications* (Springer, Berlin, 2006), pp. 287–296.
38. J. Grim, P. Somol, J. Novovičová, P. Pudil and F. Ferri, Initializing normal mixtures of densities, in *Proc. 14th Int. Conf. Pattern Recognition (ICPR'98)* eds. A. K. Jain, S. Venkatesh and B. C. Lovell (IEEE Computer Society Los Alamitos, 1998), pp. 886–890.
39. J. Grim, P. Somol and P. Pudil, Digital image forgery detection by local statistical models, in *Proc. 2010 Sixth Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, eds. I. Echizen *et al.* (IEEE Computer Society, Los Alamitos, 2010), pp. 579–582.
40. J. Grim, P. Somol, P. Pudil, I. Mikova and M. Malec, Texture oriented image inpainting based on local statistical model, in *Proc. 10th IASTED Int. Conf.* Vol. 623 (2008), pp. 15–20.
41. M. Haindl, J. Grim, P. Pudil and M. Kudo, A hybrid BTF model based on Gaussian mixtures, *Proc. 4th Int. Workshop Texture Analysis*, eds. M. Chantler and O. Drbohlav (IEEE, Los Alamitos, 2005), pp. 95–100.
42. V. Hasselblad, Estimation of parameters for a mixture of normal distributions, *Technometrics* **8** (1966) 431–444.
43. V. Hasselblad, Estimation of finite mixtures of distributions from the exponential family, *J. Am. Stat. Assoc.* **58** (1969) 1459–1471.
44. M. Heath *et al.*, Current state of the digital database for screening mammography, in *Digital Mammography* (Kluwer Academic, 1998), pp. 457–460, http://marathon.csee.usf.edu/Mammography/Database.html.
45. D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory* (Wiley, New York, 1949).
46. D. W. Hosmer Jr., A comparison of iterative maximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample, *Biometrics* (1973) 761–770.
47. O. K. Isaenko and K. I. Urbakh, Decomposition of probability distribution mixtures into their components, in *Theory of Probability, Mathematical Statistics and Theoretical Cybernetics*, Vol. 13 (VINITI, Moscow, 1976) (in Russian).
48. A. Juan and E. Vidal, Bernoulli mixture models for binary images, in *Proc. 17th Int. Conf. Pattern Recognition (ICPR'04)* (2004), pp. 367–370.
49. J. Liang, C. Xu, Z. Feng and X. Ma, Hidden Markov model decision forest for dynamic facial expression recognition, *Int. J. Pattern Recognit. Artif. Intell.* **29**(07) (2015) 1556010.
50. Y. Liu, J. Chen, C. Shan, Z. Su and P. Cai, A hierarchical regression approach for unconstrained face analysis, *Int. J. Pattern Recognit. Artif. Intell.* **29**(08) (2015) 1556011.
51. D. Lowd and P. Domingos, Naive Bayes models for probability estimation, in *Proc. 22nd Int. Conf. Machine Learning* (ACM, 2005), pp. 529–536.
52. S. C. Markley and D. J. Miller, Joint parsimonious modeling and model order selection for multivariate Gaussian mixtures, *IEEE J. Sel. Top. Signal Process.* **4**(3) (2010) 548–559.

53. G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Vol. 382 (John Wiley and Sons, New York, 2007).
54. G. J. McLachlan and D. Peel, *Finite Mixture Models* (John Wiley and Sons, New York, 2000).
55. X. L. Meng and D. Van Dyk, The EM algorithm — An old folk-song sung to a fast new tune, *J. R. Stat. Soc. B* **59**(3) (1997) 511–567.
56. E. Parzen, On estimation of a probability density function and its mode, *Ann. Math. Stat.* **33** (1962) 1065–1076.
57. C. Pearson, Contributions to the mathematical theory of evolution: 1. Dissection of frequency curves, *Philos. Trans. R. Soc. Lond.* **185** (1894) 71–110.
58. B. C. Peters and H. F. Walker, An iterative procedure for obtaining maximum-likelihood estimates of the parameters for a mixture of normal distributions, *SIAM J. Appl. Math.* **35**(2) (1978) 362–378.
59. P. Pudil, J. Novovičová, N. Choakjarernwanit and J. Kittler, Feature selection based on the approximation of class densities by finite mixtures of special type, *Pattern Recognit.* **28**(9) (1995) 1389–1398.
60. M. Saeed and N. Edu, Bernoulli mixture models for markov blanket filtering and classification, *J. Mach. Learn. Res., Workshop Conf. Proc.* **3** (2008) 77–91.
61. M. I. Schlesinger, Relation between learning and self learning in pattern recognition, *Kibernetika (Kiev)* **6**(2) (1968) 81–88.
62. M. I. Schlesinger and V. Hlaváč, *Ten Lectures on Statistical and Structural Pattern Recognition*, Vol. 24 (Springer Science and Business Media, 2013).
63. J. Šochman and J. Matas, WaldBoost — Learning for time constrained sequential detection, in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR 2005)*, IEEE Computer Society, 2005), pp. 20–25.
64. R. L. Streit and T. E. Luginbuhl, Maximum likelihood training of probabilistic neural networks, *IEEE Trans. Neural Netw.* **5**(5) (1994) 764–783.
65. D. Tran and M. Wagner, A fuzzy approach to speaker verification, *Int. J. Pattern Recognition Artif. Intell.* **16**(7) (2002) 913–925.
66. I. Vajda, *Theory of Statistical Inference and Information* (Kluwer Academic, Dordrecht, 1989).
67. I. Vajda and J. Grim, About the maximum information and maximum likelihood principles in neural networks, *Kybernetika* **34**(4) (1998) 485–494.
68. S. Vajda, T. Plötz and G. A. Fink, Camera-based whiteboard reading for understanding mind maps, *Int. J. Pattern Recognit. Artif. Intell.* **29**(03) (2015) 1553003.
69. J. H. Wolfe, Pattern clustering by multivariate mixture analysis, *Multivariate Behav. Res.* **5** (1970) 329–350.
70. X. Zhou, X. Xing, L. Han, H. Hong, K. Bian and K. Xie, Structure feature learning method for incomplete data, *Int. J. Pattern Recognit. Artif. Intell.* **30**(09) (2016) 1660007.

Jiří Grim graduated in Physical Electronics (M.S.-level) from the Czech Technical University, Prague. He received his Ph.D. in Computer Science from the Czech Academy of Sciences, Prague, in 1981. Currently he is with the Institute of Information Theory and Automation of the Czech Academy of Sciences, Prague. His research interests include application of distribution mixtures to statistical pattern recognition, neural networks, machine learning, statistical modeling, expert systems, image analysis and others.